SLNet 5.0

Installation, Configuration, Administration, and User Guide

Version 5.0

High Performance Telnet Server for Windows

SLNet 5.0

Installation, Configuration, Administration and User Guide

August 2010

Published by:

SeatlleLab

www.SeattleLab.com

© 1996-2010 SeattleLab. All Rights Reserved.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of SeattleLab.

SLNet®, SLNetRF®, and SeattleLab® are trademarks of SeattleLab. Other brands or products are the trademarks or registered trademarks of their respective holders and should be treated as such.

Any errors, corrections, suggestions, or omissions should be sent to documentation@seattlelab.com.

SLNET-5.0 ICAU-UD5-IN1

Table of Contents

Introduction **1**
Full-Featured Telnet Service 1
Telnet Security 1
Hand Held RF Device Support 1
SAP Console compatibility 1

Architecture **2**

SLNet Requirements **2**
Adobe Reader or Acrobat 2

Upgrading from a Previous Release **2**

SLNet Features **3**
High Performance 3
Access to Windows's Common Command Shell 3
Direct Access to Custom Applications 3
Security 3
Secure Logon 3
Terminal Emulation 3
Logging 3
Remote Administration 3
Disconnects 3

What's New in the SLNet 5.0 Release **4**
Licensing 4
Windows 2008 support 4
Improved Performance and Reliability 4
Improved Admin Interface 4
More Parameters 4
Unicode and UTF-8 support 4
Broadcast Messages 4
User Status 4
Extended Logging and Playback 4
Multi-layer configuration – user, terminal, system 4
Menus and Customizations 4
Security 4
Manage Multiple SLNet Servers from a single interface 4

SLNet and Windows Security **5**
Authentication types 5

SSH Security **6**
Handheld device security 6
Secure Shell 6

Installation of SLNet 5.0 7

Configuration 8

Service 9
Server Name: 9
User Connections: 9
Logging: 10
Logon Banner: 11
Status: 11
Stop, Start or Pause SLNet: 11
Alternative Ways to Start and Stop SLNet 11
Console 12
Auto Reconnect: 13
User Prompts: 14
Default User: 14
Auto Logon IPs 15
Enable Auto Logon 15
Add a single IP address and User ID. 16
Add IP addresses using a mask. 16
Add a range of IP addresses and User ID. 17
Edit an auto logon entry. 17
Delete an auto logon entry. 17
Terminal Types 18
Override Service and Console settings 19
User Redirection Characters: 20
Users 22
Override settings 23
Default Shell: 24
Default Start Application: 24
Use Redirection Characters: 24
Include / Exclude 27
Connected Users 28
Send a message 28
License 29
Get a License Key 29
Load a License Key 31
About 32

Logging on to SLNet 33

Logon Welcome Message 33
User Prompts 33
Automatic Logon 34
Auto Logon by IP address 34
Exiting an SLNet session 34
Automatic Reconnect 34

SLNet Terminal Emulation 35

SLNet Logging **32**
Logging Message examples 32
SLNet Error Codes **33**
Appendix A. Troubleshooting **34**
Before You Contact SLNet Technical Support 34
SeattleLab Professional Services 34
Contacting SLNet Specialists 34
Appendix B. Customization **35**
Named Pipes 35
Appendix C. Miscellaneous **37**
SLclient32 37
Secure Logon Capabilities 37
Appendix D. Multiple Console Windows **38**
Select Console Type 38
SLCons and SLCon 38
Working with Multiple Console Windows 39
Using the Menu 40
Locating and dispatching Popups 41
Hot Key Commands 41

This page intentionally left blank.

Introduction

SLNet is the most efficient, most flexible and best value telnet server solution available. The architecture, performance and features are robust enough for large enterprises, yet affordable and flexible enough for any small business.

SLNet telnet server runs on any Windows 2000, 2003, or 2008 server. SLNet enables an unlimited number of users to work invisibly and simultaneously on any Windows server without interfering with one another or the server console user. Additionally, SLNet optimizes the telnet protocol between Radio Frequency (RF) devices and Microsoft Windows servers.

Full-Featured Telnet Service

- Allows **unlimited concurrent connections** for any device running terminal emulation, including RF devices.
- Users can be placed **directly** into custom applications or access the Common Command Shell.
- Use **existing** character-based and DOS applications to run multi-users without modification.
- Written in open Internet standards to work with any telnet client. **No proprietary** client software is required.
- Remote users can work invisibly and simultaneously on the same Windows 2000, 2003, or 2008 server without interfering with the server console user.
- Features two levels of 'time-out' including: the **clean-up** of processes when a connection is dropped and **inactivity time-outs** to log clients off after a specified period of no activity.
- System administrators can include more information such as security warnings for logon banners through the multi-line banner support.
- Gives administrators the ability to **customize the logon and password prompts**.
- Allows administrators to **send messages** to logged on users without interrupting their work flow.

Telnet Security

- Completely **preserves** Windows security protocols.
- Requires user authentication by the **Windows security** authority.
- Gives administrators the ability to configure SLNet **permissions** on a user-by-user basis.
- **Restricts** logon rights to users with "log-on locally" permissions.
- Gives users the same permissions as if they were working at the console.
- Allow only **specified** users to log on via Telnet.
- **Restricts** logon rights to specific IP addresses or address ranges.
- Allows the logon sequence to be **encrypted**

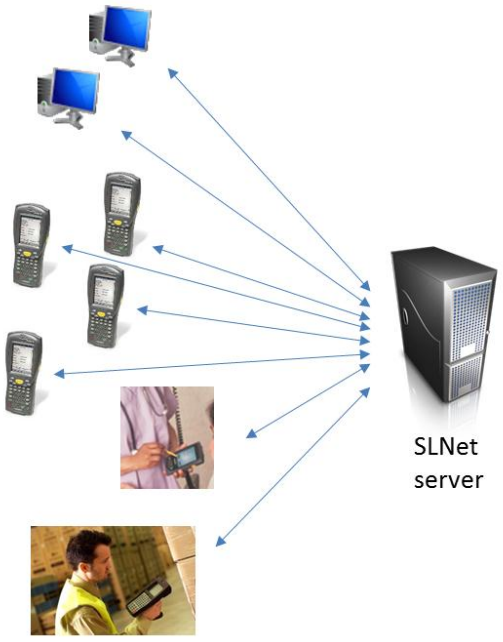
Hand Held RF Device Support

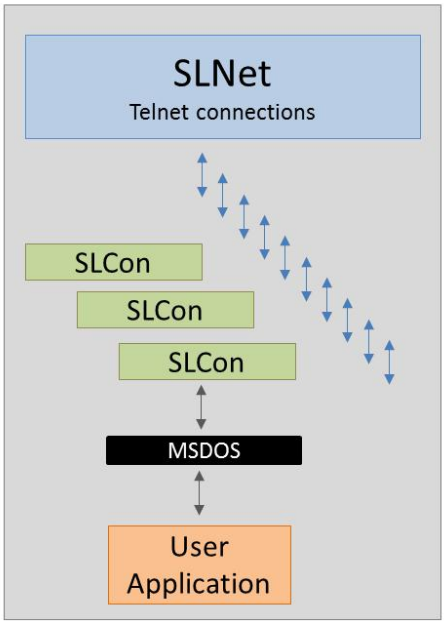
- SLNet allows **unlimited, concurrent RF devices** connections.
- Ready for real-time use with any wireless radio frequency (RF) device that supports the telnet protocol.
- Users can access Windows 2000, 2003, or 2008 hosts from a device.
- Support small screen sizes.
- **No extensive application rewrites** or the investment of learning new and cumbersome wireless APIs.

SAP Console compatibility

SLNet is the standard for interfacing to the SAP Console from RF devices. It is in use at major organizations worldwide. They chose SLNet because of its outstanding performance and reliability. It installs in a few minutes and delivers a powerful solution for organizations from a few people to thousands.

Architecture





The SLNet server handles a large number of connections (thousands) by handling all connections, terminal emulation and character translation through an SLCon.exe for each user that connects up to the user's application through an MS-DOS prompt.

SLNet Requirements

SLNet 5.0 supports Windows Server 2000, 2003, and 2008 as a SLNet Server.

The SLNet service itself uses only a minimal amount of system resources, requiring four megabytes of disk space in your root system directory (the directory where Windows is installed on your system) and 2 megabytes of system RAM for the initial connection.

Adobe Reader or Acrobat

You will need to install a copy of Adobe Reader to view the help documentation (PDF format). You can download it for **free** at http://www.adobe.com.

Upgrading from a Previous Release

SLNet has been a standard in the computing industry for over a decade. It is the most widely used Telnet solution.

If you are a previous SLNet user of 2.6 or 4.1 versions, you should upgrade to the 5.0 release for improved operation and security, additional platform support, and new features.

You will need to uninstall the previous SLNet version and install the new 5.0 release.

SLNet Features

High Performance
SLNet communicates via Winsock Sockets to the terminals and via Pipes to SLCon and the MSDos Command Prompts. To minimize impact on the CPU resources, the SLNet programs are heavily multi-threaded and asynchronous.

Access to Windows's Common Command Shell
Users who Telnet into SLNet can have full access to Windows' Common Command Shell. SLNet is completely configurable by the administrator, who can choose to disable access to the Command Shell on a user-by-user basis.

Direct Access to Custom Applications
SLNet offers you the power to define default custom applications for users as soon as they log in.

Security
Users run in their own appropriate security context, with all their own permissions in place, just as if they had logged in from the main console. This completely preserves all Windows's C2 security features.

Secure Logon
SLNet encrypts user IDs and passwords for secure logon over the Internet. With SLNet's AuthTypes registry value you can specify the level of encryption (and the encryption protocol) for the entire system, or on a user-by-user basis.

Terminal Emulation
SLNet provides ANSI terminal sequences (compatible with the VT-100 and VT-220) for cursor positioning and formatted screen handling. It supports colors, UTF-8 characters, Unicode characters and custom key sequences.

Logging
SLNet can insert a record in the Windows event log every time a user connects or disconnects along with a record of what process the user executed and the resulting data and screens from that process.

Remote Administration
SLNet can be administered locally or remotely.

Disconnects
In a ***controlled*** disconnect, a telnet packet from the terminal lets SLNet know the terminal is about to disconnect. This allows SLNet to tell the various threads that the events they are waiting for will not be occurring.

In an ***uncontrolled*** disconnect, the threads will either go into a temporary wait mode or exit cleanly and close the sockets, pipes, and MSDOS Command prompt and applications as needed.

A terminal user or device can reconnect securely to a temporarily disconnected application.

What's New in the SLNet 5.0 Release

Licensing
For security purposes SLNet is licensed to specific servers. A license key tied to the server ensures you can track your SLNet licenses and prevent unauthorized uses. Install SLNet on a target server, it activates temporarily and provides a license request file that you email to the SeattleLab license group. A license key file is generated and sent back to you. It easily installs into your SLNet system to authorize that server for the licensed time period and the number of concurrent users.

Windows 2008 support
SLNet fully supports Windows 2008 servers in 32 bit and 64 bit deployments.

Improved Performance and Reliability
SLNet can handle thousands of user connections with modest requirements.

Improved Admin Interface
The SLNet configuration interface has been reworked and improved for ease of use and more flexibility. For larger organizations, it is easier to configure users.

More Parameters
Additional parameters have been added for customization of users, terminal types, and system settings. SLNet is more flexible and offers powerful choices without complexity to make any telnet environment connect to applications smoothly and reliably.

Unicode and UTF-8 support
SLNet now has full support for international languages – single and double byte character sets. You can set the font and code page to meet any need.

Broadcast Messages
SLNet can broadcast any message to any number of users without interrupting their work flow.

User Status
The SLNet console shows all connected users and their status and latest activity.

Extended Logging and Playback
New SLNet logging has been added for everything a Telnet administrator would need. This includes a full playback capability where you can see every keyboard action and all internal SLNet processes. Resolving any technical issue is now simple and easy.

Multi-layer configuration – user, terminal, system
SLNet offers a powerful way to manage customizations by passing up settings from system to terminal to users. This allows customizations to be done at a high level and automatically passed through to users.

Menus and Customizations
SLNet can be easily customized to offer your users menus to choose domains or applications.

Security
SLNet can lock down terminals so they can only reach specific applications and be prevented from getting to a command prompt.

Manage Multiple SLNet Servers from a single interface
The SLNet Configuration panel has a server selection. You can select any SLNet server on your network and change its configuration settings.

SLNet and Windows Security

SLNet abides by all Windows security features. It does not alter nor change any security settings or privilege levels set by normal Windows mechanisms.

When a Telnet user logs into your system via SLNet, he or she must be a valid user that you or another administrator created using the Windows User Manager or Active Directory Manager. SLNet sets that user's process to run in the security context appropriate to that user. The user then has exactly the same privileges he or she would have on your machine if they sat down in front of it and logged in on the console by pressing the `CTRL ALT DELETE` keys.

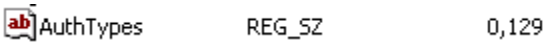
Windows allows you to grant or withhold privileges on a user-by-user basis to every object on the system. Using Windows Explorer, for example, you may choose which individual users can have access to any given file you create. SLNet abides by these exact same access restrictions. It does not and cannot alter any privileges set by Windows.

Authentication types

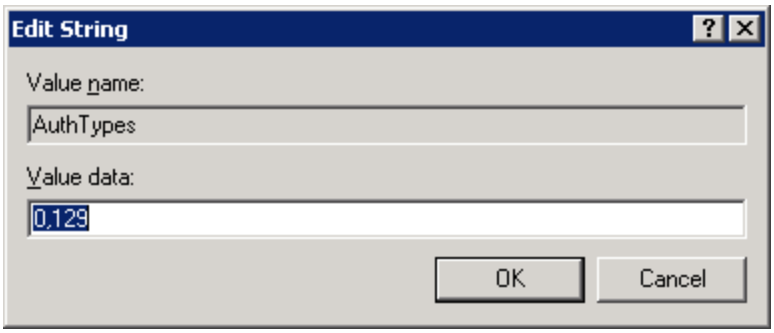
SLNet allows you to encrypt the user name and password in the data stream with the **AuthTypes** registry value.

You will need to modify this registry value in the root of the SLNet registry `(HKEY_LOCAL_MACHINE > SOFTWARE > SeattleLab > SLNet)`. Open the Registry Editor through the Start Menu (Start > Run and then entering `REGEDIT`).

In the right frame of the registry editor, right-click on the **AuthTypes** string value and select Modify from the popup menu.



By default the key has 0,129 (IP Address, decimal equivalent for the **AuthType** required), which allows all users to use the SeattleLab encryption or fall back to clear text.



Enter `<IP address, decimal equivalent for the authorization type; IP address,...>` and then click OK to update the string. The string pairs must be separated by a semi colon. For example: `0,129;127.0.0.1,128;129.168.0.1,65`

The authentication types are:

- Null (no encryption)—bit 1, decimal 1
- Microsoft NTLM—bit 7, decimal 64
- SeattleLab proprietary—bit 8, decimal 128

Note: Multiple encryption types can be specified by combining the bits. For example for Null (clear text) and SeattleLab encryption the decimal equivalent would be 129. All single encryption types and pairs are possible.

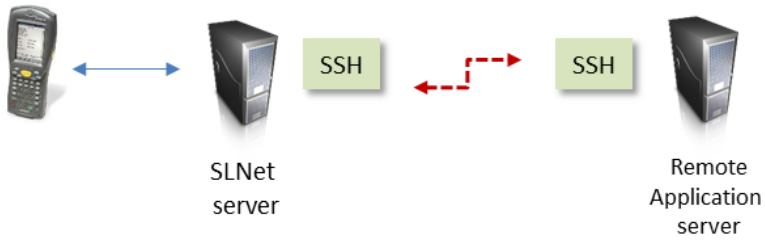
SSH Security

Handheld device security

The wireless data transmitted between the RF Guns and the Base Station is usually encrypted at the network level using WEP, WPA, WPA-PSK, WPA-PSK2, etc.

If your applications are on remote servers outside of your network, you may want to set up a secure connection for the communication between the SLNet server and your application servers. The entire SLNet data stream can be “tunneled” through a Secure Shell encrypted link using “port forwarding”. *All the traffic between the connecting machine and the destination host will be encrypted.*

There are a number of commercial and free open source Secure Shell servers and clients available.



Secure Shell

A Secure Shell (SSH) tunnel consists of an encrypted tunnel created through an SSH protocol connection. To set up an SSH tunnel, you configure an SSH client to forward a specified local port to a port on the remote machine. Once the SSH tunnel has been established, the user can connect to the specified local port to access the network service. The local port need not have the same port number as the remote port.

You can set up an SSH tunnel with the SLNet Server. It is recommended that you install the SSH server software on the same machine with the SLNet Server.

The SSH Client gets installed on a remote network or network segment where the SLNet Clients are located.

The SSH Client acts as a port forwarder. It can be configured to accept connections on port 23 (the Telnet port). It will forward the data for those connections over the secure SSH tunnel to the SSH Server.

The SSH server forwards the data to its local SLNet Server. The SSH server returns data from the SLNet Server over the SSH tunnel to the SSH Client. The SSH Client will then send that data to the appropriate SLNet Client.

The SSH servers need to have a User configured authorize a connection from the SSH Client. The User needs to be authorized to do Port Forwarding.

The SSH Clients need to be configured to allow them to connect to the SSH Server (SSH Server authorized Username and Password information). They need to be configured to tell the SSH Server how to forward data from the SSH Client and also to monitor a network port for connections from SLNet Clients.

The syntax for setting up the port forwarding on the SSH Client is usually 3 parameters:

1. Local Port to monitor for SLNet Clients (this may consist of an IP address followed by colon (:) and then the Port, flag characters that are shortcuts for the IP followed immediately by the Port or just the Port (with an implied local or all local IPs).
2. The IP address and Port of the SLNet Server relative to the IP of the SSH Server. If the SSH Server is on the same machine as the SLNet Server, the IP of the SLNet Server would be 127.0.0.1 (the local IP). The Port of the SLNet Server will be 23 (unless it was changed).
3. The IP (and optionally the port) of the SSH Server relative to the SSH Client. Some SSH Clients do not use a 3rd parameter. They require the IP and Port of the SSH Server to be specified separately.

Installation of SLNet 5.0

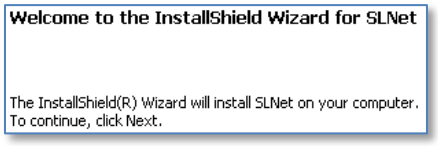
Before installing SLNet 5.0, you must first log on as a member of the Administrator's user group.

Download the SLNet 5.0 installation file from the SeattleLab web site or obtain it from a reseller.

Put it on the local drive on the Windows server.

Double click it to start the installation process.

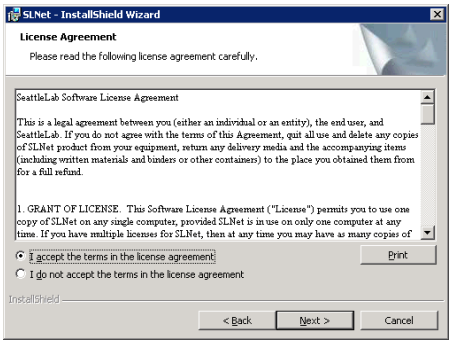
Click ***Next*** to continue.


Welcome to the InstallShield Wizard for SLNet
The InstallShield(R) Wizard will install SLNet on your computer. To continue, click Next.

Read the License Agreement.

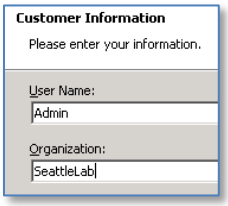
Check "I accept the terms in the license agreement."

Click ***Next*** to continue.


SLNet - InstallShield Wizard
License Agreement
Please read the following license agreement carefully.
SeattleLab Software License Agreement
This is a legal agreement between you (either an individual or an entity), the end user, and SeattleLab. If you do not agree with the terms of this Agreement, quit all use and delete any copies of SLNet product from your equipment, return any delivery media and the accompanying items (including written materials and binders or other containers) to the place you obtained them from for a full refund.
1. GRANT OF LICENSE. This Software License Agreement ("License") permits you to use one copy of SLNet on any single computer, provided SLNet is in use on only one computer at any time. If you have multiple licenses for SLNet, then at any time you may have as many copies of
I accept the terms in the license agreement
I do not accept the terms in the license agreement
Print
InstallShield
< Back
Next >
Cancel

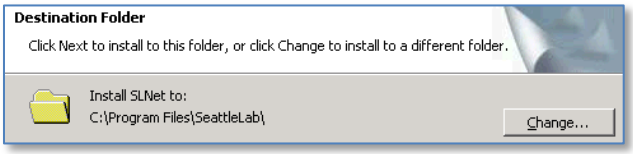
Enter a User Name and Organization.

Click ***Next*** to continue.

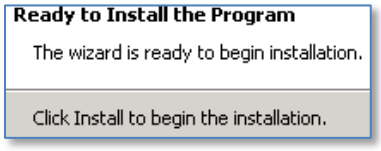


Change the destination folder for the SLNet software if needed.

Click ***Next*** to continue.

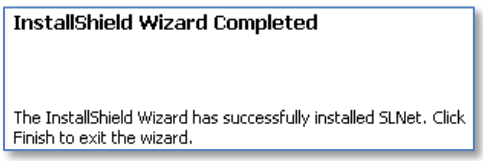

Destination Folder
Click Next to install to this folder, or click Change to install to a different folder.
Install SLNet to:
C:\Program Files\SeattleLab\
Change...

Click ***Install*** to continue.



Installation of SLNet is complete.

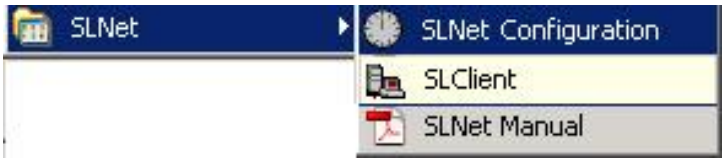
Click ***Finish***.


InstallShield Wizard Completed
The InstallShield Wizard has successfully installed SLNet. Click Finish to exit the wizard.

Configuration

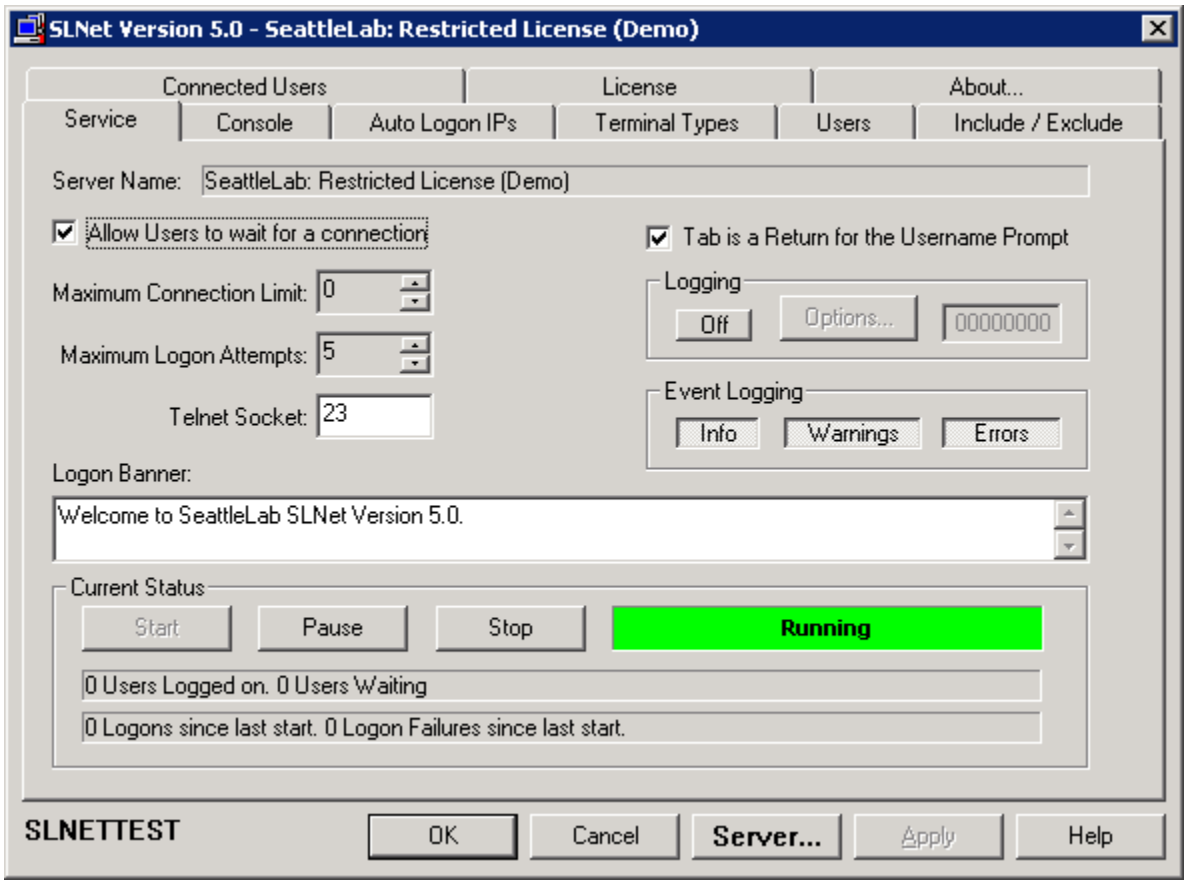
SLNet offers a powerful selection of options for customization of your Telnet needs.

Go to Start programs and select SLNet Configuration.



The SLNet configuration panel is displayed.

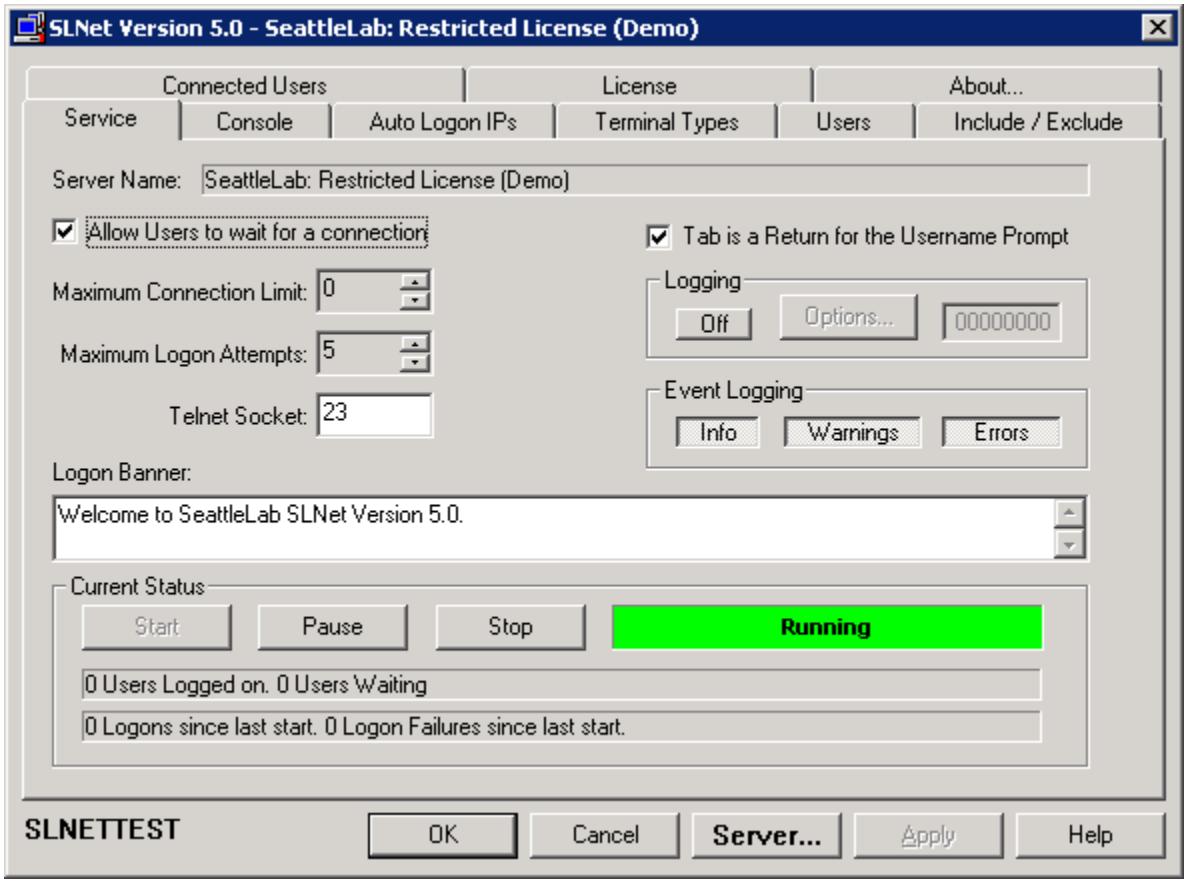
Until you enter a valid license key, SLNet is in a demonstration mode with a temporary license key that expires.



If you see the License Key expires warning prompt, click ***OK*** to continue.

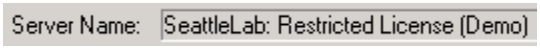
If you have a License Key, click on the License tab to load your valid SLNet License Key file.

Service



Server Name:

The Server Name is a description of the server that SLNet is running on. It is a combination of your organization name and the registered server / location name that you supplied to generate a license key.

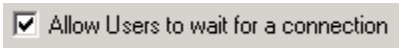


This is the default Server Name for an unlicensed server. SLNet is installed automatically as a Demonstration license with a limited number of users and a time expiration.

You can change to a different server by clicking on the **Server…** button and entering a different server name. The server must be connected to your network. The server's computer name appears on the left bottom of the panel.

User Connections:

You can allow users to wait for a connection if all licenses are in use.



If this box is **Unchecked**, the users will see a message,

```
There are currently too many Telnet
sessions. Please try again later.
```

They will then be returned to the Telnet prompt.

Your license will specify the maximum number of user connections allowed.

You can change it to a lower number if needed.

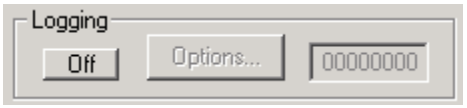
You can set the security limit for logon attempts.

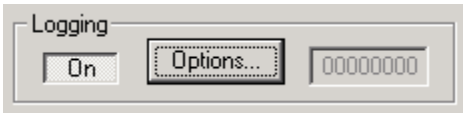
Specify the Telnet socket (port) that SLNet will monitor for connections. *Changes you make to the Telnet Socket in the Service Options do not take effect until you restart the SLNet service.*

Miscellaneous: The tab key can be set as a return for the user name prompt.

Logging:

Logging is **Off** by default. *Logging should be left **Off** on production systems unless directed by a SeattleLab technician.*





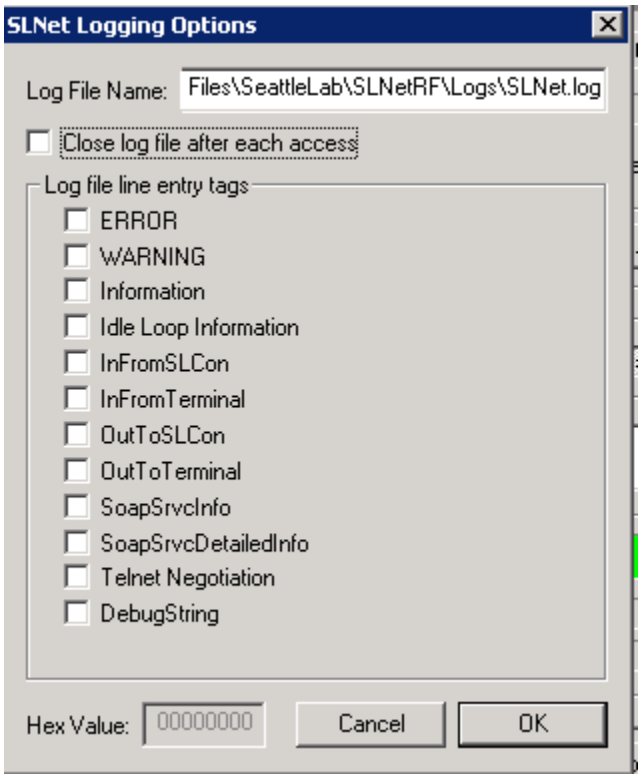
Click the Logging button to turn **On** logging.

Select **Options** for logging parameters.

The total number of log entries is displayed.

SLNet uses the Windows Application Log in the Event Viewer to keep a record of everything it does by entering messages as events occur.

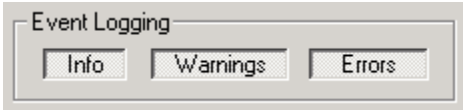
If you set Logging On, you can set which events you want logged.



Check the box next to an item to log its information.

These settings are usually used by SLNet technical support staff to assist you in resolving issues.

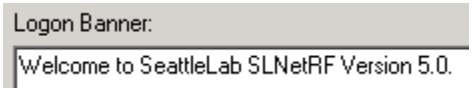
By default, SLNet logs three types of events: Info, Warnings and Errors.
Info messages (such as which user logged on) are the least serious and Errors (such as your demo copy of SLNet has expired) are the most serious.



You can turn off the recording of Events. Toggle each button to turn off its Event recording.

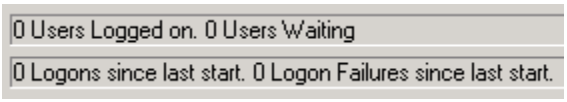
Logon Banner:

You can change the welcome message sent to each device when it connects to SLNet - *maximum number of characters is 2048*.



Status:

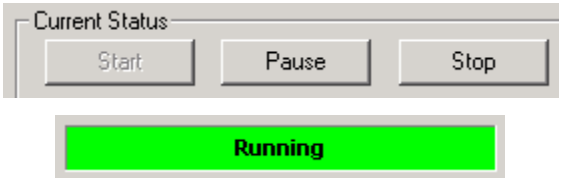
SLNet tracks current active users and any users waiting for connections (may need to increase your licensed amount of users).



Stop, Start or Pause SLNet:

You can Pause, Stop or Start SLNet.

If you stop the SLNet service, the service is completely inactivated. It is removed from memory and is no longer listening on the Telnet socket. Telnet users who attempt to connect to your Windows system will receive a connection refusal message.



Alternative Ways to Start and Stop SLNet

The SLNet service can also be controlled from the Services applet in the system control panel. In the Services scroll box, choose SeattleLab Telnet Server. You may then start, stop or pause the service.

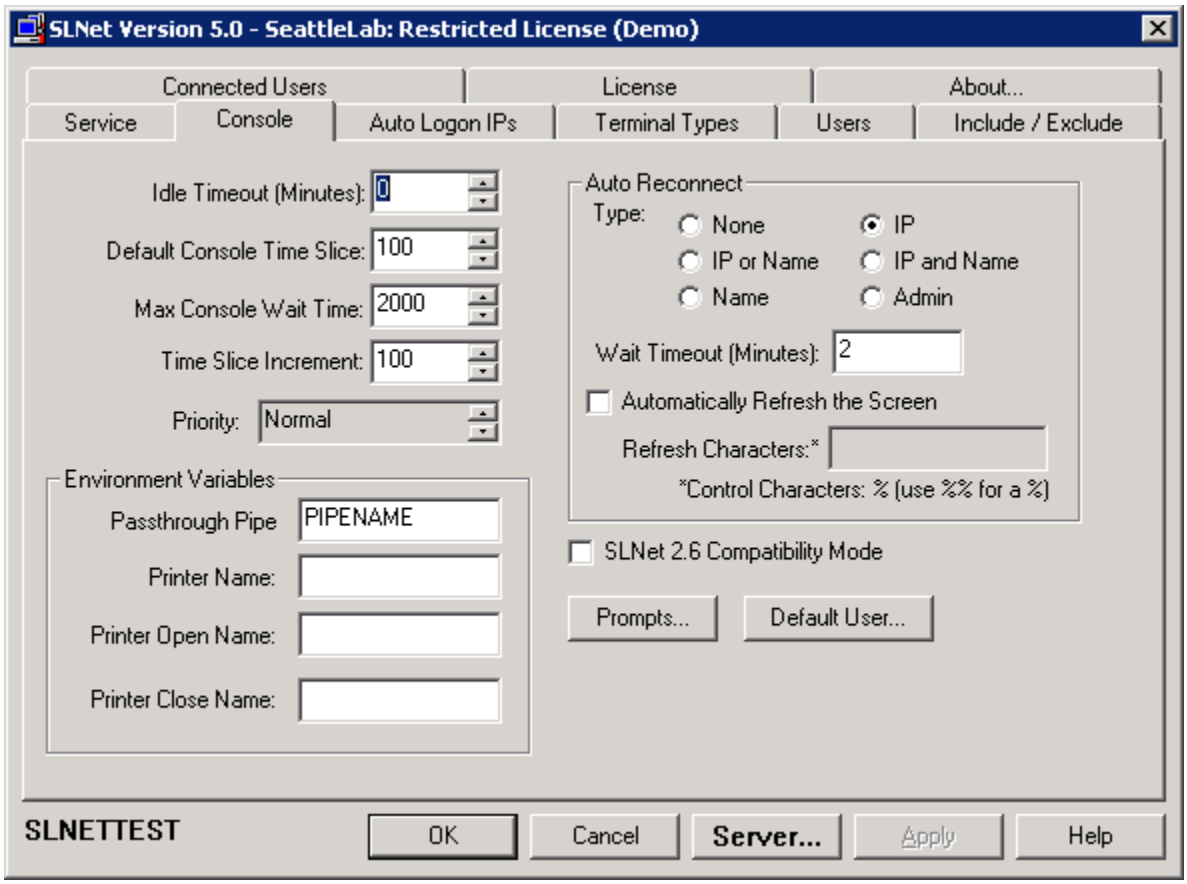
SLNet can also be started or stopped directly from the Windows Common Command Shell.

```
net start slnet
net stop slnet
```

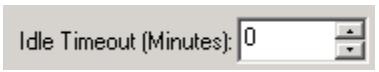
Console

The Console tab of the SLNet Configuration Control Panel allows you to set internal parameters used in the virtual consoles created by the SLNet service.

Do not adjust any of these parameters without advice from a trained SeattleLab technician.
Doing so will change performance, latency, and throughput of SLNet and your Windows system as a whole.

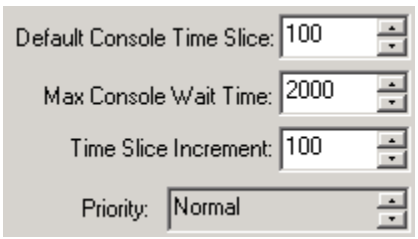


You may set Idle Timeout for as many minutes as you choose (0 value means **do not** timeout). If there is no activity from a user in the amount of time entered, SLNet will close the process.



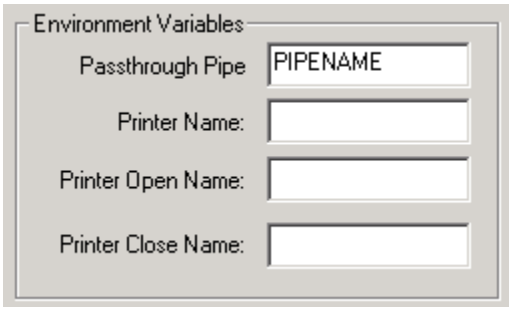
Suppose you cannot back up your system if there are processes running. If your users have a habit of not logging off at the end of the day, you can set this option in order to end the processes so that you can back up your system.

Only adjust these settings with the help of a trained SeattleLab technician.



The SLNet service monitors the screen buffer of a user's console for changes, so that those changes may be sent over Telnet to update the screen of the client. If SLNet monitors these changes too infrequently, the user will experience sluggish and choppy performance. If SLNet monitors those changes too frequently, it will consume too much of the system's resources and reduce the performance of the SLNet user's task and all other tasks running on the system. The algorithm used by SLNet to attempt to optimize these conflicting constraints is to begin polling relatively frequently, and to poll progressively less frequently when no screen activity is

detected.



Only adjust these settings with the help of a trained SeattleLab technician.

The **Passthrough Pipe** allows you to customize the environment variable name of the main pipe used by SLNet. The default name for this field is PIPENAME.

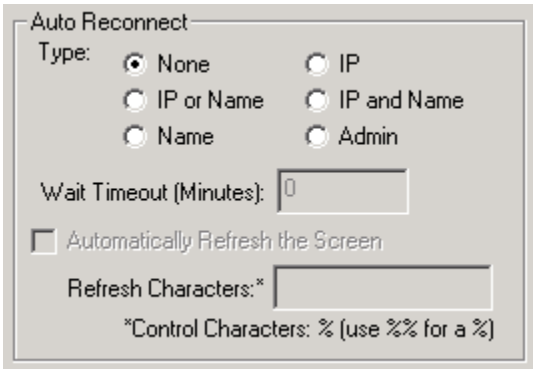
The **Printer Name** is the environment variable to contain the printer name associated with the user or terminal.

The **Printer Open Name** is the environment variable to contain the Escape sequence that enables the user printer.

The **Printer Close Name** is the environment variable to contain the Escape sequence that disables the user printer.

See Appendix B. for customization options.

Auto Reconnect:



SLNet can be configured to automatically reconnect to devices that get disconnected. Set the Auto Reconnect values:

Security - Reconnect only if match on

- None – *no auto reconnect*
- IP address or User Name
- User Name
- IP address
- IP address and User Name
- Admin user

Wait Timeout – keep connection available – then exit and clean up.

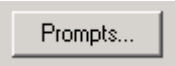
To refresh the screen upon a reconnect enter the character string to send to the application.

- %R will cause SLNet to redraw the screen from its last saved screen buffer in SLCon.
- You can enter any command that will cause the user's application to refresh the screen.

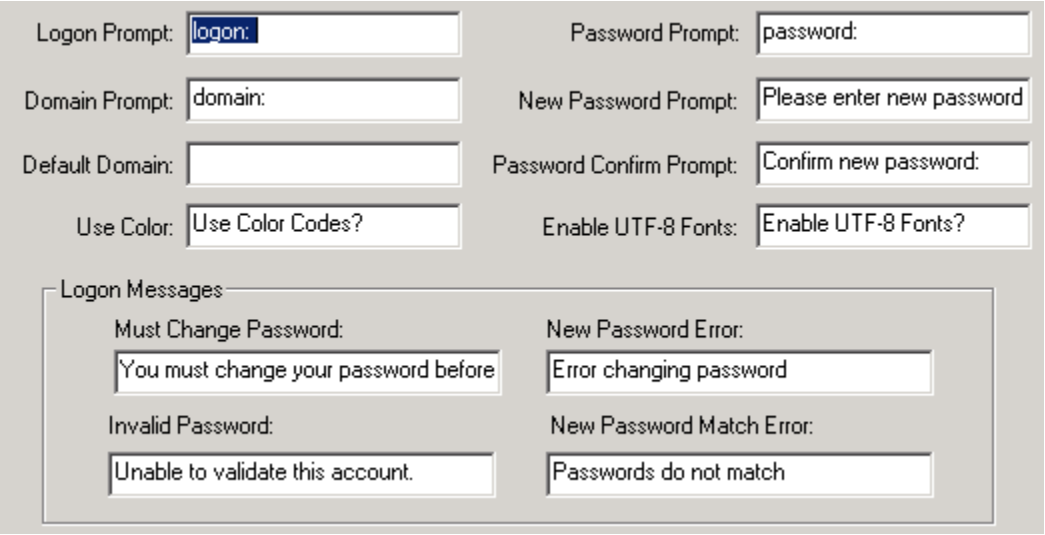
Optional setting for older SLNet installations
(not usually required for application compatibility).

User Prompts:

You can change any of the SLNet prompts.



If you enter a Windows domain name in the Default Domain box, users will not see a domain prompt and SLNet will access the default domain automatically.

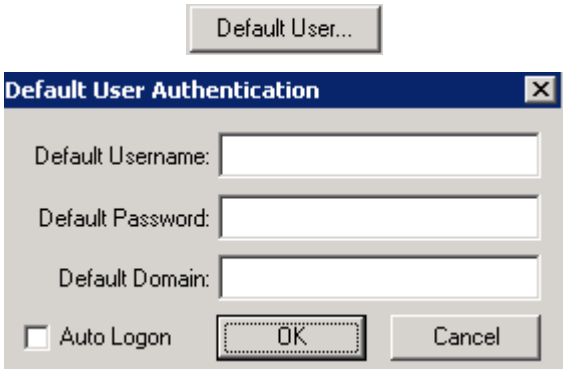


These prompts can be changed to any text (language independent).

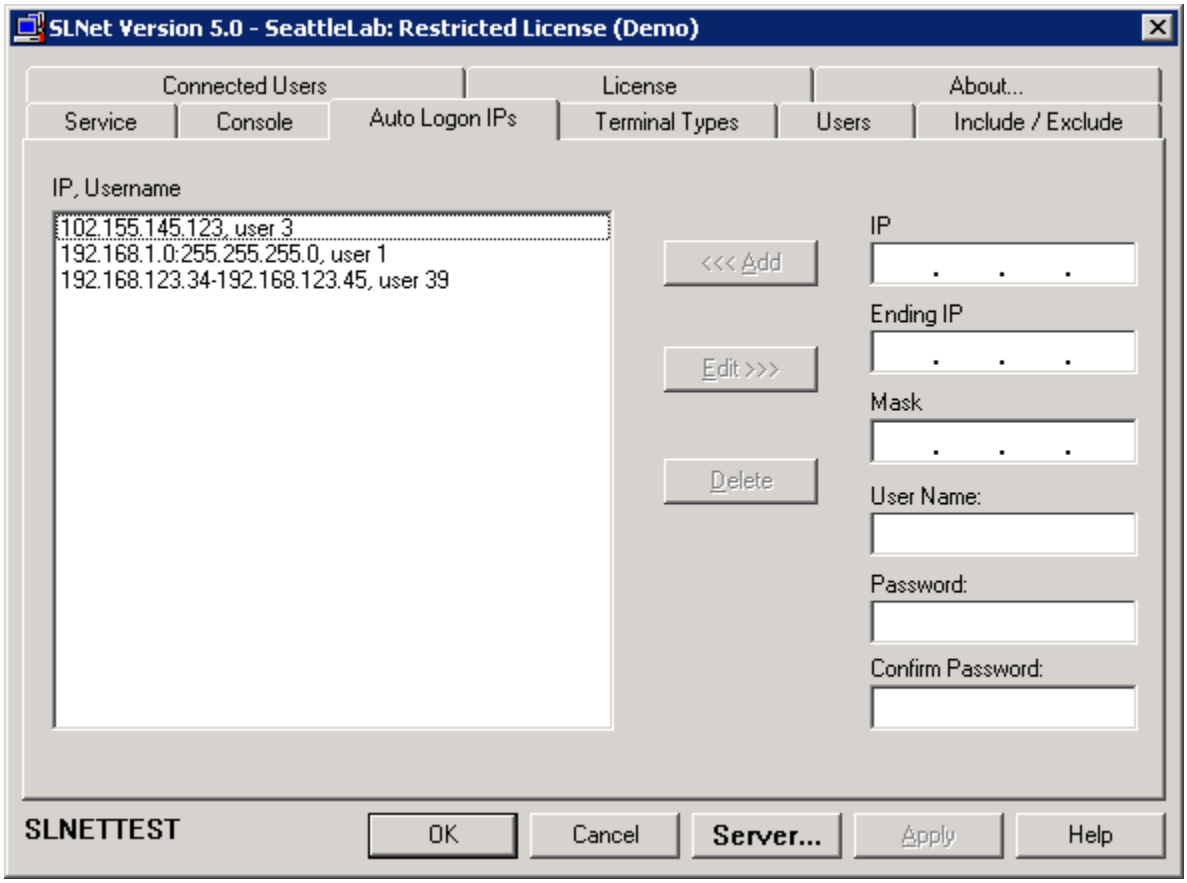
Default User:

If you set terminals to Auto Logon, then you need to set the Default User Logon and Password.

All terminals will be logged on as this User.



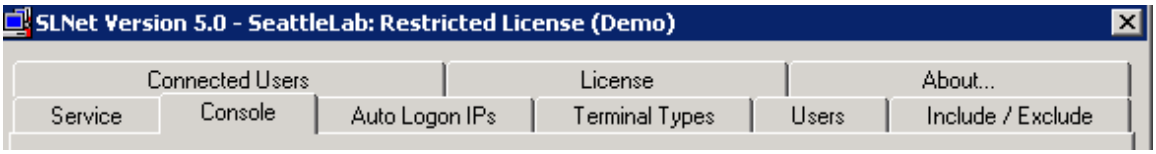
Auto Logon IPs



SLNet can automatically logon a user from a device that connects from a specific IP address or range of addresses*.*

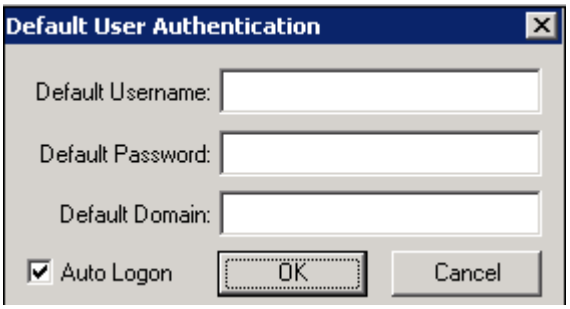
This enables you to have a shared user ID be used from a single IP address or a secured range of IP addresses.

Enable Auto Logon



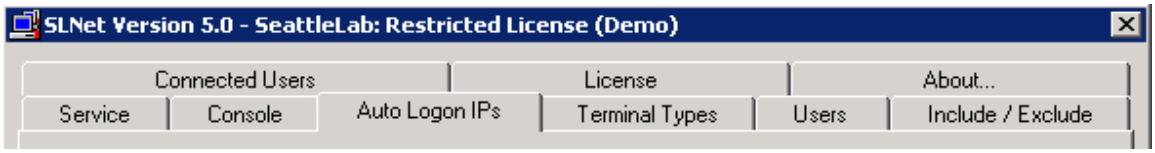
On the Console tab, click on the ***Default User…*** button.

Do not enter any information for the Default Username, Password or Domain.



Check the **Auto Logon** box, click ***OK***.

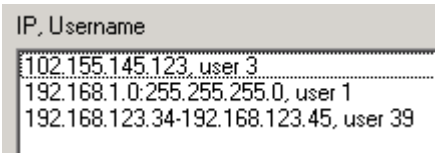
Select the Auto Logon IPs tab.



Use a **single IP address** to log on with a specific user ID.

The format is **`IPaddress,userID`**.

If you need a IP Mask, the format will be **`IPaddress:Mask,userID.`**



Use a **range of IP addresses** to log on with a specific user ID.

The format is **`IPaddress-IPaddress,userID`**.

Add a single IP address and User ID.

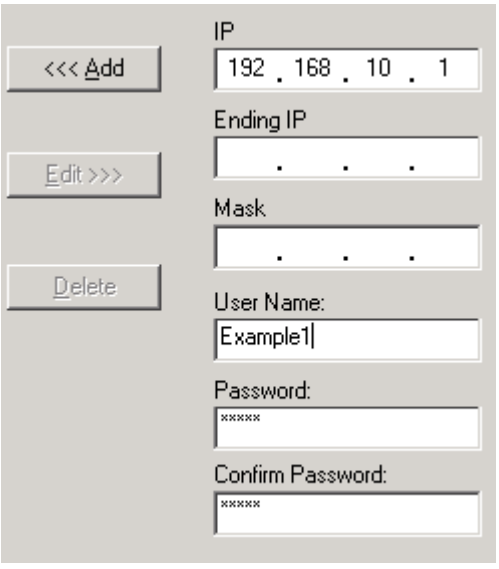
Enter the **IP address**.

Enter a **User Name**.

Enter a **Password** (not required).

Confirm the Password.

Click the ***<<<Add*** button.



Add IP addresses using a mask.

Enter the starting **IP address**.

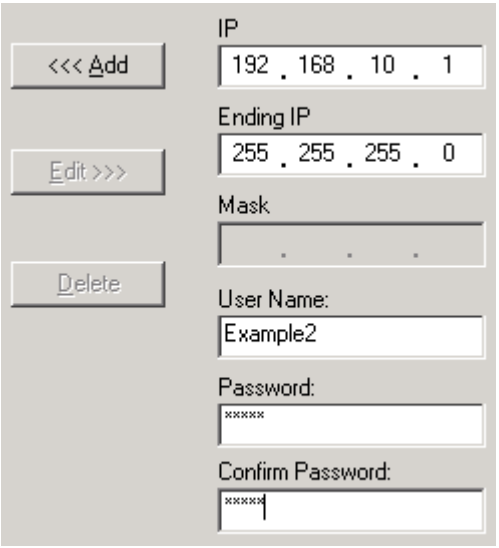
Enter the **mask** to populate the IP address range.

Enter a **User Name**.

Enter a **Password** (not required).

Confirm the Password.

Click the ***<<<Add*** button.



Add a range of IP addresses and User ID.

Enter the beginning **IP address**.

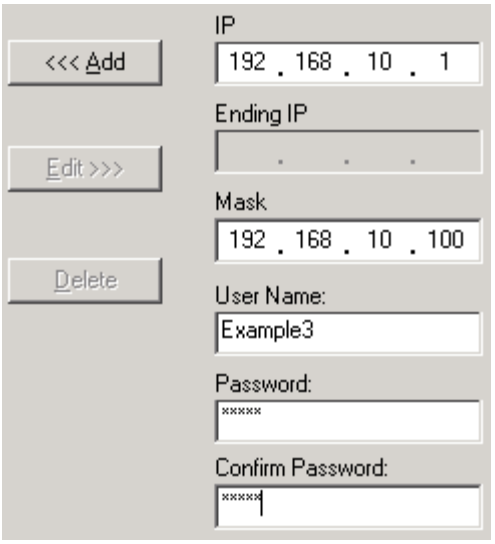
Enter the ending **IP address**.

Enter a **User Name**.

Enter a **Password** (not required).

Confirm the Password.

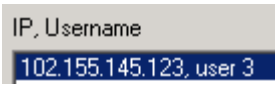
Click the ***<<<Add*** button.



Edit an auto logon entry.

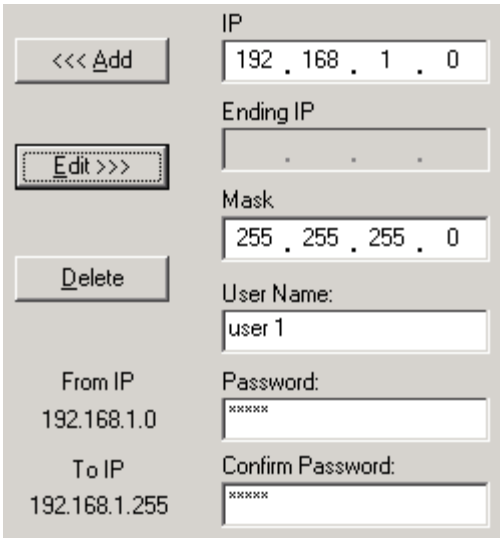
Highlight **IP address**.

Click the ***<<<Edit*** button.



Make any changes.

Click the ***<<<Add*** button.

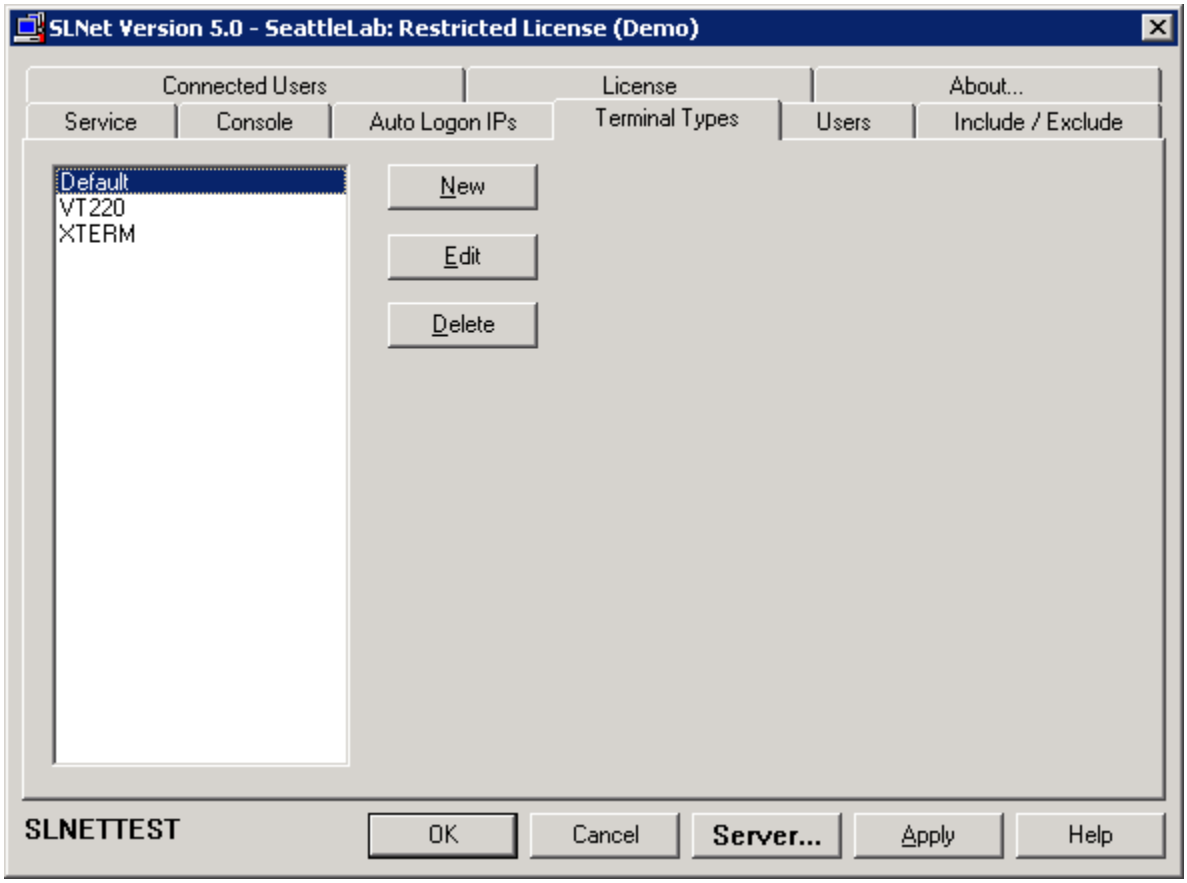


Delete an auto logon entry.

Highlight **IP address**.

Click the ***Delete*** button.

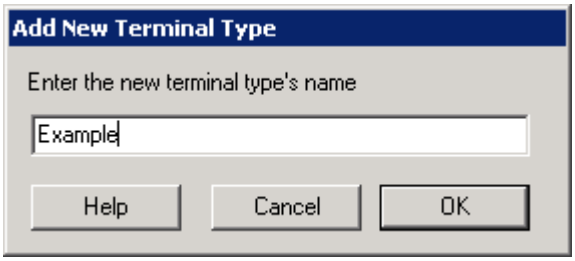
Terminal Types

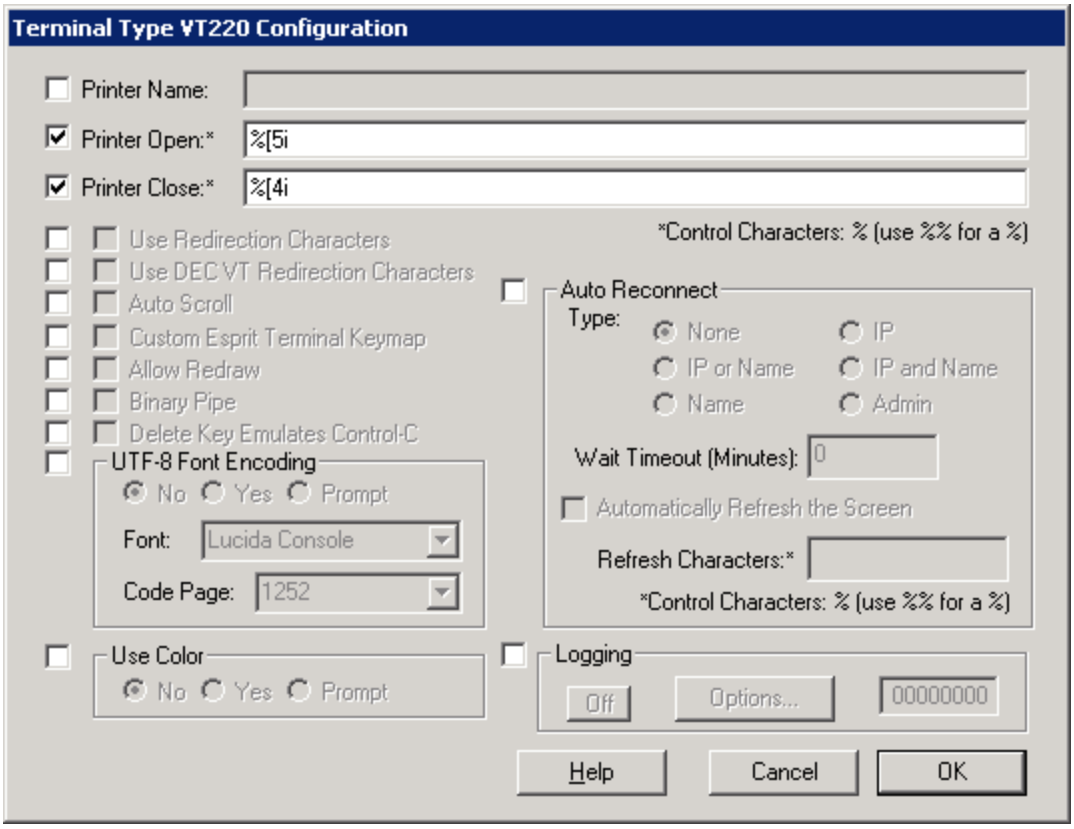


SLNet can support an unlimited number of terminal types. You can set a default terminal type or enter any new terminal types that you need. ***It must match the ID String from the terminal or Port Server.***

The information entered in the terminal settings is passed to the user.

Select a terminal type and click **Edit** to make changes or **Delete** to remove a terminal type from the list. Click **New** to add a new terminal type.





Override Service and Console settings

The flexibility of SLNet allows you to override the default Service and Console settings. The two columns of check boxes allow you to override a setting's value and then be able to activate it.

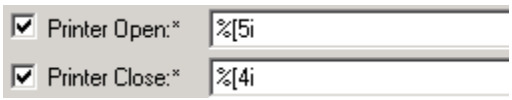
Not checked and grayed out – use Service or Console setting.

Checked and not grayed out – setting can be modified but is **disabled**.

Checked and Checked – setting can be modified and setting is **enabled**.

Set the default printer for this terminal type. Check the box to enter the UNC or IP address.

Enter the command string to open the printer or close the printer.



User Redirection Characters:

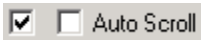
This option (default is enabled) determines whether non-printable characters (characters less than 32 decimal) are mapped to printable characters. The 32 non-printable characters will be converted to printable characters based on the following table. When a non-printable character is encountered, the printable equivalent will be substituted in its place. If this option is disabled, all non-printable characters will be represented with spaces.

Decimal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Non-Printable Character	☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
RNT Equivalent	@	*	n	*	*	$	*	o	*	o	o	o	!	|	*

Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Non-Printable Character	►	◄	↕	‼	¶	§	■	↨	↑	↓	→	←	∟	↔	▲	▼
RNT Equivalent	>	<		#	#	$	-	|	^	v	>	<	-	%	^	v

Select this option to provide DEC VT emulation.

This option is defaulted to **disabled**.



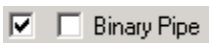
If the Auto Scroll option is enabled, when a character is output to the last position on the screen (the 80th column of the 25th row), the screen scrolls up.

Most Telnet clients support auto scrolling. If the Telnet client is set to perform scrolling, the Auto Scroll option should be disabled in SLNet to prevent incorrect screen formatting.

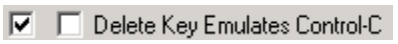
This option is a specific customization for an Esprit terminal.

This option (default is enabled) clears the screen if you switch from screen scraping mode to using a custom name pipe.

If this option is enabled, all translation of characters in the input stream is disabled. This would be appropriate, for example, for users writing custom communications software such as a file transfer protocol.

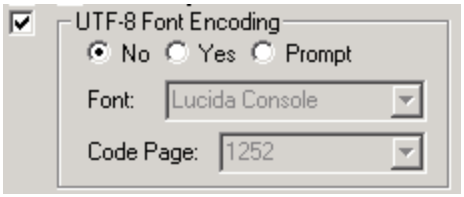


You can customize the Delete key to send a CTRL-C to the application.



Enable UTF-8 fonts. Select a font and a Windows code page to use.

This allows UniCode characters to display on UTF-8 compatible terminals.



Set the Auto Reconnect values:

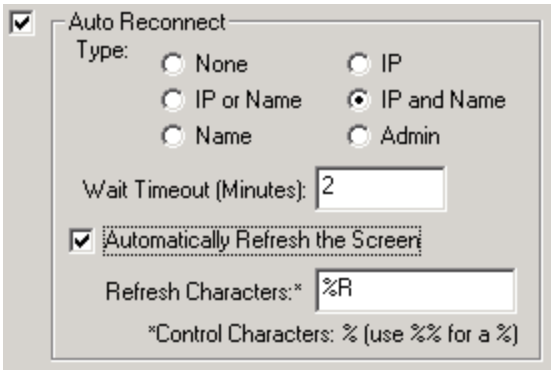
Security - Reconnect only if match on

- None – *no auto reconnect*
- IP address or User Name
- User Name
- IP address
- IP address and User Name
- Admin user

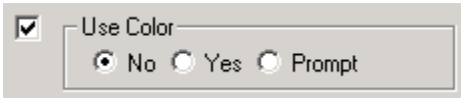
Wait Timeout – keep connection available – then exit and clean up.

To refresh the screen upon a reconnect enter the character string to send to the application.

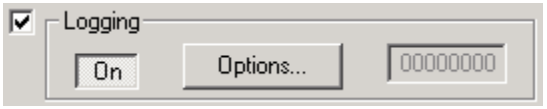
- %R will cause SLNet to redraw the screen from its last saved screen buffer in SLCon.
- You can enter any command that will cause the user's application to refresh the screen.



Activate color.



Set logging on – for all sessions using this terminal type.

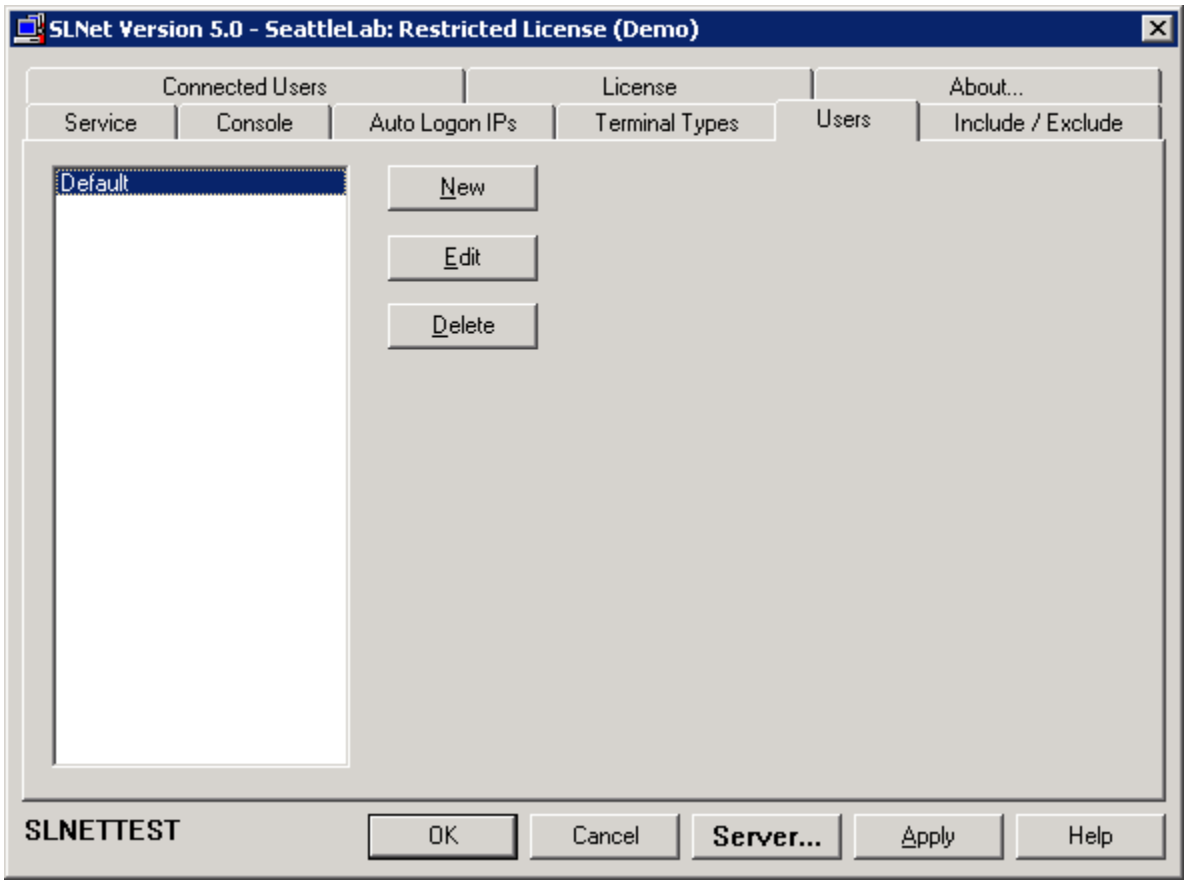


Users

You can add as many custom user profiles to SLNet as needed.

All SLNet users must have an existing Windows user account.

SLNet allows you to define system access on a user-by-user basis. It also allows you to automatically customize the start for each user - for example, all users are presented with the Windows Common Command Shell or SAP Console is started.

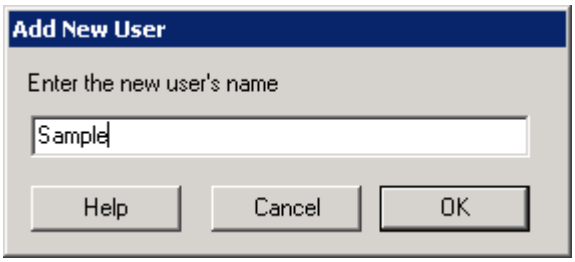


This is **Default** configuration is used for any user who logs onto SLNet but does not have a custom profile. Most organizations use the **Default** user.

Warning! If the Default configuration is ever removed, only individual users who have been specifically added to the SLNet user list will be allowed to log on via SLNet.

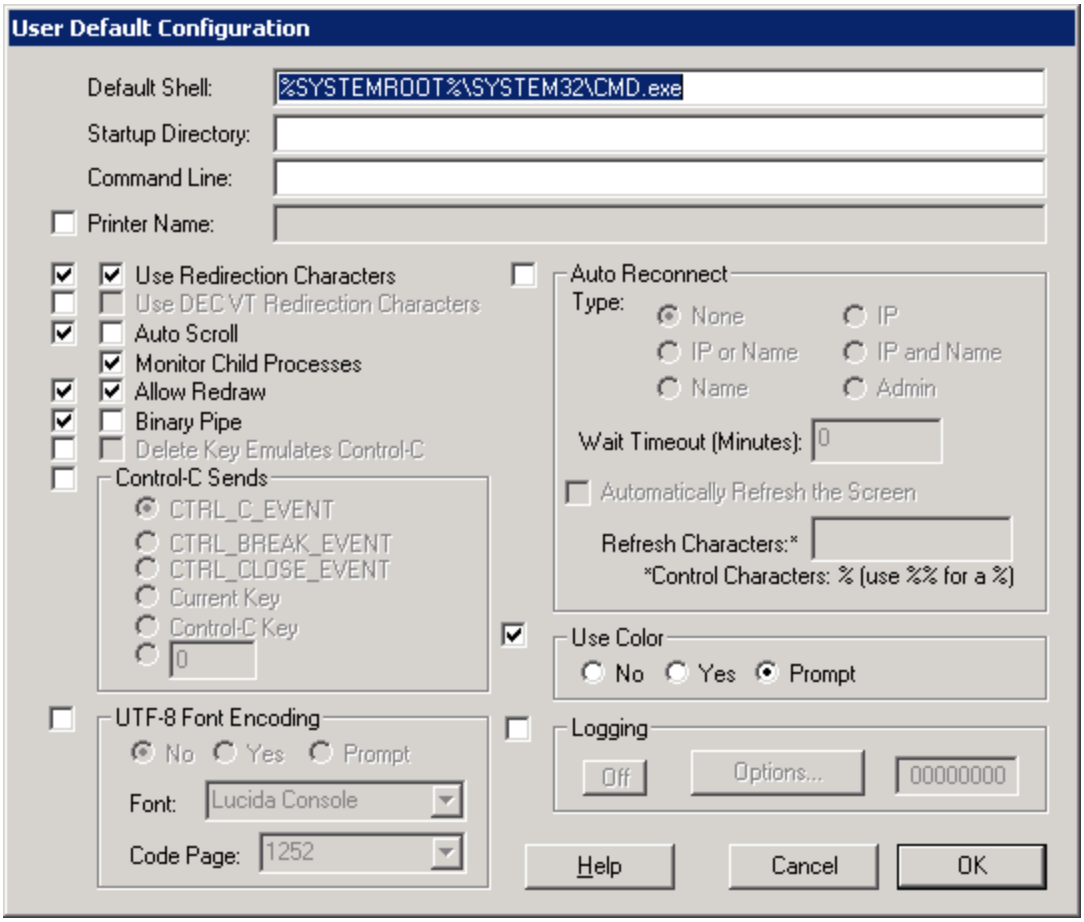
Select a user type and click **Edit** to make changes or **Delete** to remove a user from the list.

Click **New** to add a new user. It must be a valid Windows user account.



If Auto Logon is enabled, then all users will be logged in as the Default user.

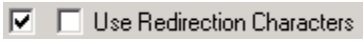
If Auto Logon is not selected, then users will be prompted for their Windows / Active Directory User ID and Password.



Override settings

The flexibility of SLNet allows you to override the default Service, Console and Terminal settings. The two columns of check boxes allow you to override a setting's value and then be able to activate it.

Not checked and grayed out – use Service or Console setting.



Checked and not grayed out – setting can be modified but is **disabled**.

Checked and Checked – setting can be modified and setting is **enabled**.

Default Shell:

The *Default Shell* field should be left as configured upon installation of SLNet. This default path allows access to the system command shell, which is the same command prompt you see from your console when you click the MSDOS icon in the Main program group. Opening the command shell allows console applications to run properly as well as giving access to the prompt in the command shell if desired.

Default Start Application:

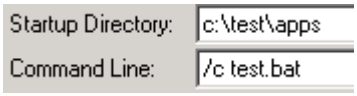
To put users directly into your own custom application, specify the directory and application startup command.

Enter the filename in the text box preceded by a switch command (`/c` or `/k`) followed by any required parameters for the application.

The complete paths *Startup Directory* fields must be entered in the format of either the Universal Naming Convention (UNC) or drive letters.

Paths entered must be LOCAL.

Example:



This is the default printer to use for this user. It will be assigned to the Printer Name environment variable defined on the Console tab.

Use Redirection Characters:

This option (default is enabled) determines whether non-printable characters (characters less than 32 decimal) are mapped to printable characters. The 32 non-printable characters will be converted to printable characters based upon the following table. When a non-printable character is encountered, the printable equivalent will be substituted in its place. If this option is disabled, all non-printable characters will be represented with spaces.

Decimal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Non-Printable Character	☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
RNT Equivalent	@	*	n	*	*	$	*	o	*	o	o	o	!	|	*

Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Non-Printable Character	►	◄	↕	‼	¶	§	■	↨	↑	↓	→	←	∟	↔	▲	▼
RNT Equivalent	>	<		#	#	$	-	|	^	v	>	<	-	%	^	v

Select this option to provide DEC VT emulation.

This option is defaulted to **disabled**.

If the Auto Scroll option is enabled, when a character is output to the last position on the screen (the 80th column of the 25th row), the screen scrolls up.

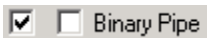
Most Telnet clients support auto scrolling. If the Telnet client is set to perform scrolling, the Auto Scroll option should be disabled in SLNet to prevent incorrect screen formatting.

When this option is enabled any child processes that are spawned by the user's primary application will also be monitored by SLNet. With this option, SLNet can insure that all user processes are properly stopped and cleaned up before exiting.

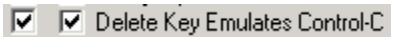
Some applications will not run if this option is enabled. If an application functions directly from the Windows system console but will not function with SLNet, disable this option.

This option (default is enabled) clears the screen if you switch from screen scraping mode to using a custom name pipe.

If this option is enabled, all translation of characters in the input stream is disabled. This would be appropriate, for example, for users writing custom communications software such as a file transfer protocol.

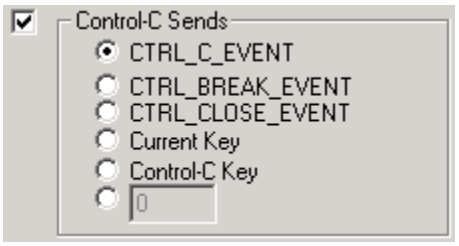


You can customize the Delete key to send a CTRL-C to the application.



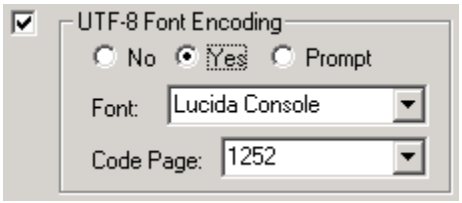
You can customize the Control –C character string passed to the application.

Make a selection or enter an ASCII value.

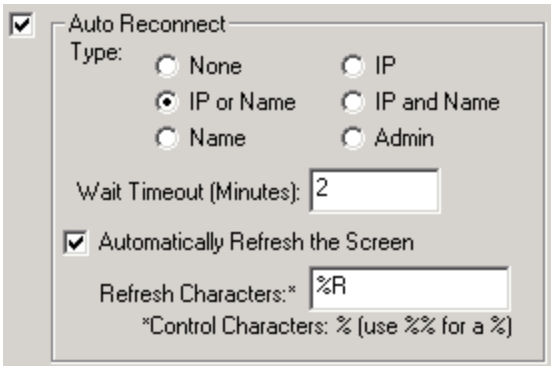


Enable UTF-8 fonts. Select a font and a Windows code page to use.

This allows UniCode characters to display on UTF-8 compatible terminals.



Set the Auto Reconnect values:



Security - Reconnect only if match on

- None – *no auto reconnect*
- IP address or User Name
- User Name
- IP address
- IP address and User Name
- Admin user

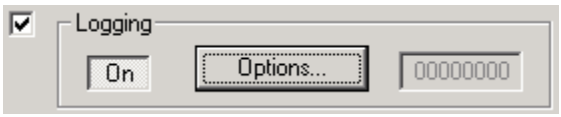
Wait Timeout – keep connection available – then exit and clean up.

To refresh the screen upon a reconnect enter the character string to send to the application.

- %R will cause SLNet to redraw the screen from its last saved screen buffer in SLCon.
- You can enter any command that will cause the user's application to refresh the screen.

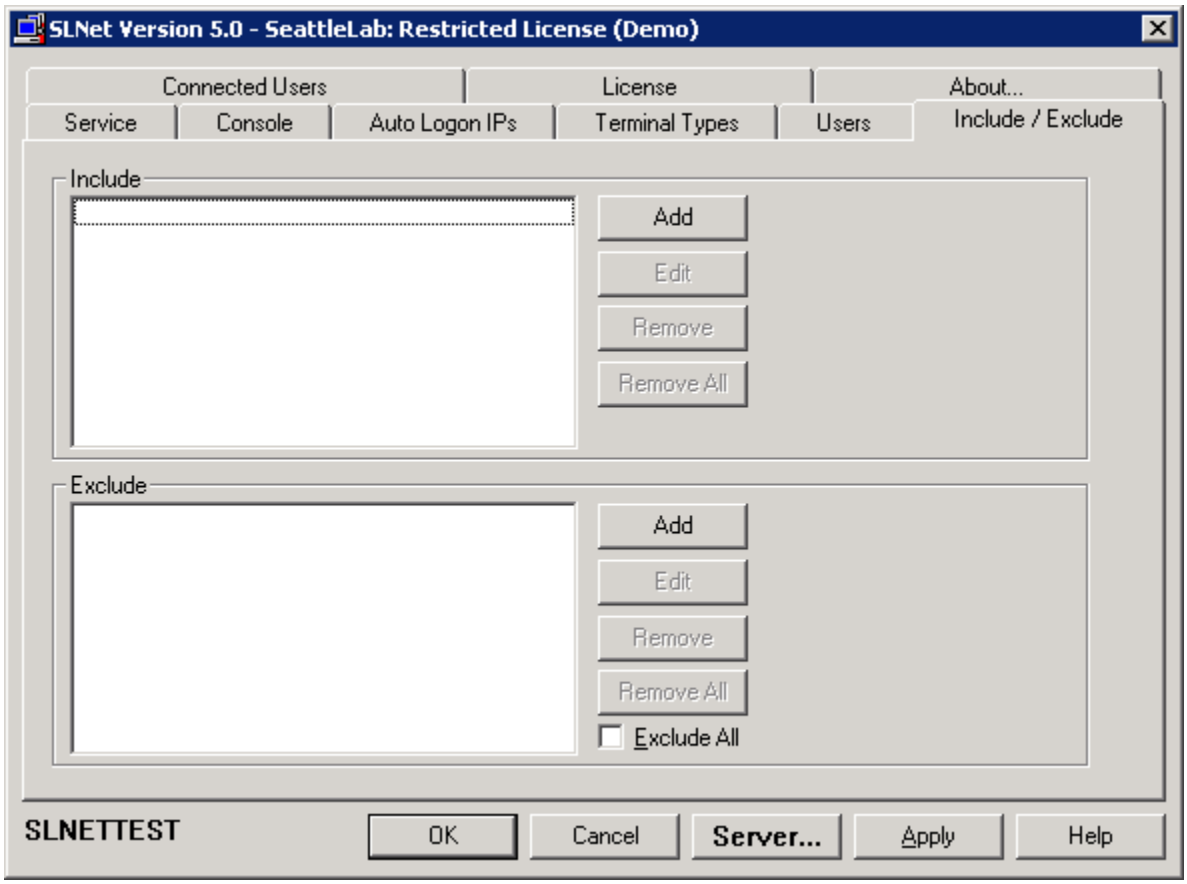
Activate color.

Set logging on for this user.



Include / Exclude

You can control the locations that can access your server – from a single device to a range of addresses.



Click **Add** to add a new device or address range.

Choose **Include** to allow access.

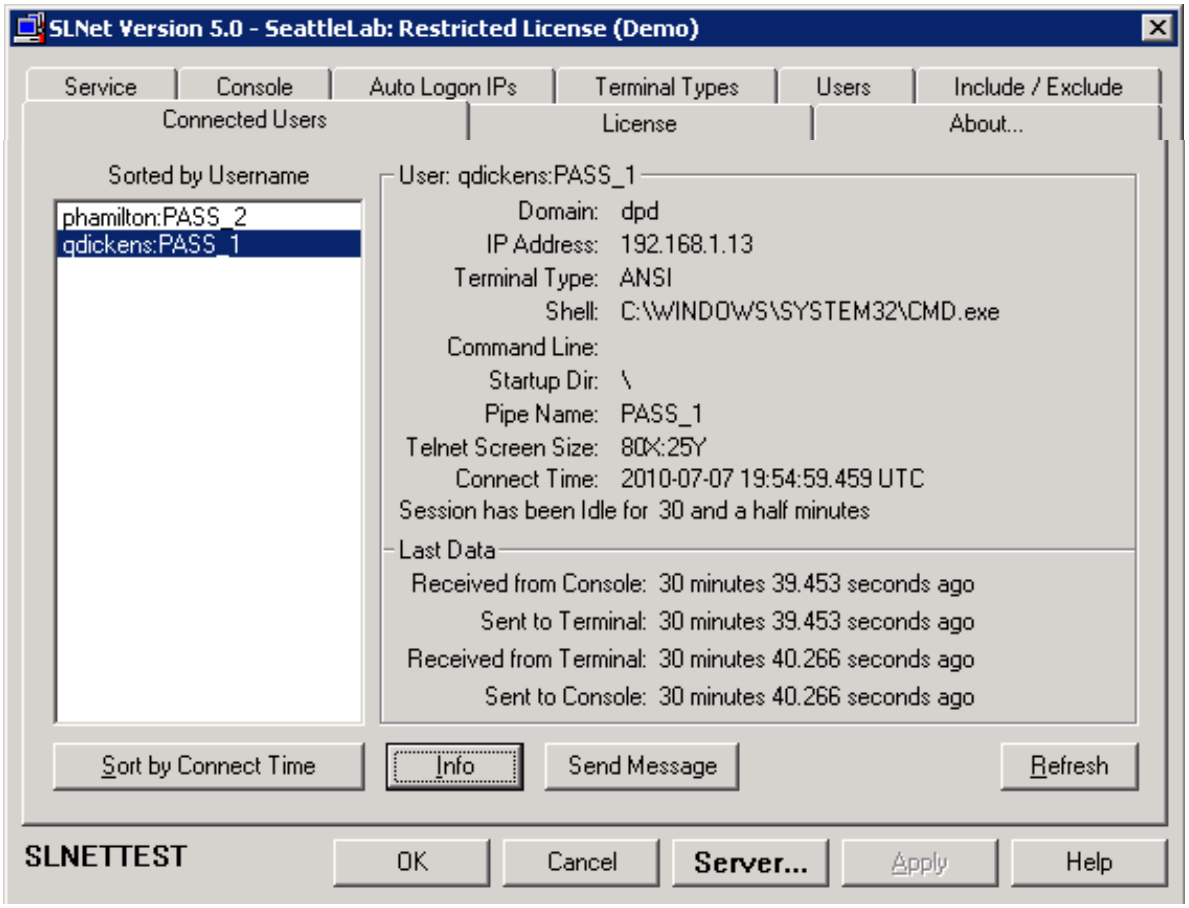
Enter an IP address or range of addresses.

Choose **Exclude** to deny access.

Enter an IP address or range of addresses.

Select a location and click **Edit** to make changes or **Remove** to remove a device or range of addresses from the list.

Connected Users



You can monitor which users are connected to your server.

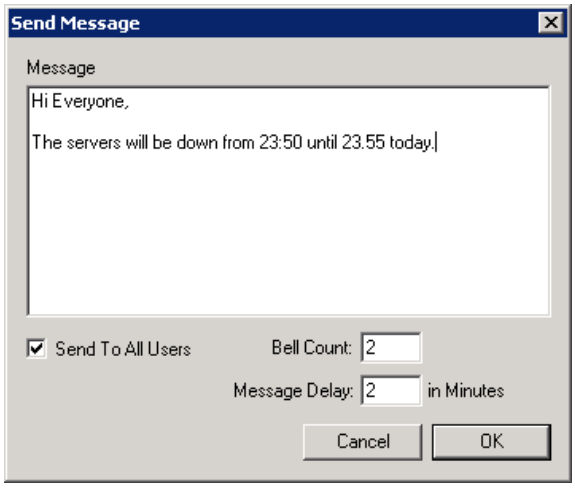
Select a user to see the details on that user. Click the ***Info*** button.

Send a message

You can send a message to any user. Select the User and click the **Send Message** button.

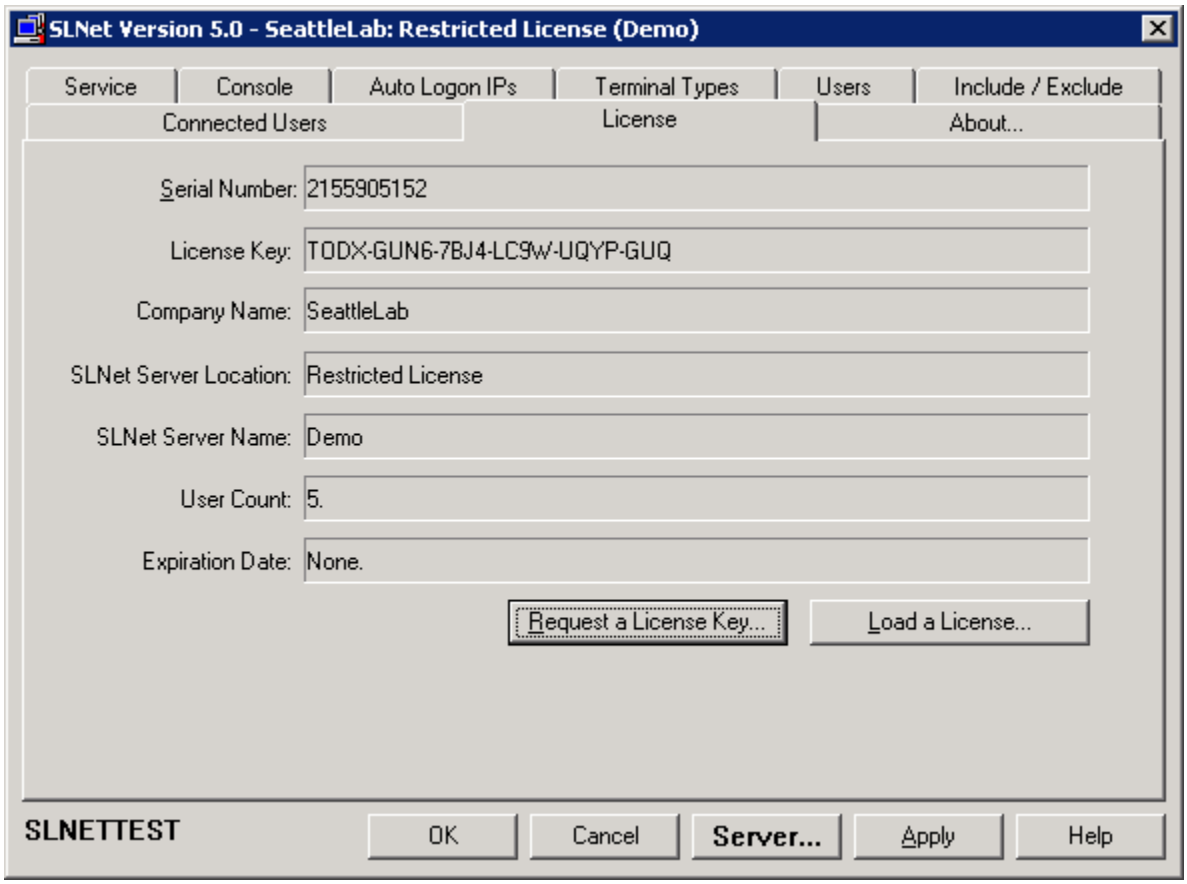
Check the box **Send to All Users** to do a broadcast to all connected users.

Broadcast messages do not interrupt the terminals input flow. The application will continue to execute. Characters sent from the terminal will go to the application even while the broadcast message is displayed.



License

SLNet is licensed by server and the number of concurrent users.



Get a License Key

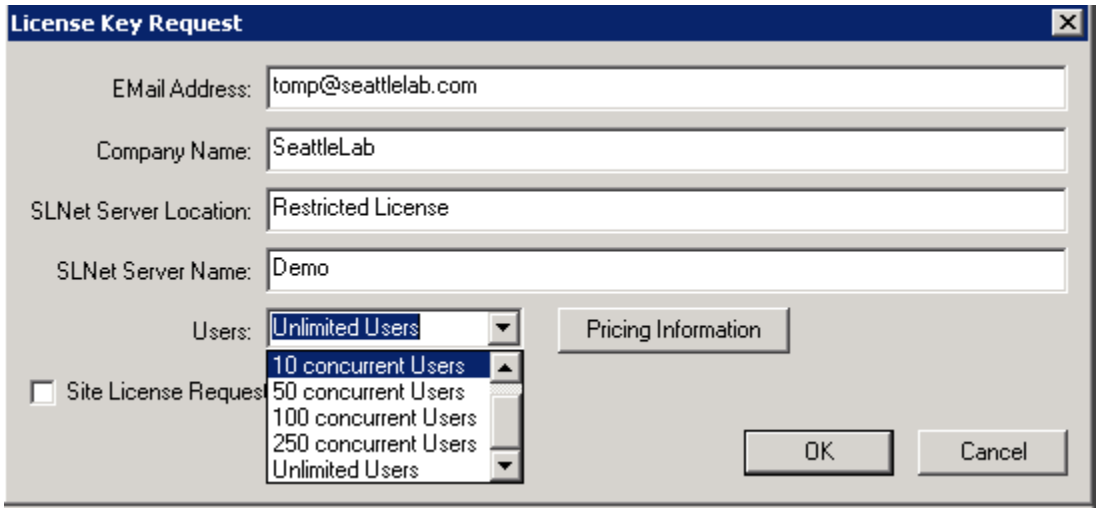
After you have installed SLNet on a server, click the ***Request a License Key*** button

to generate a file to email to the SeattleLab license department. An email will be returned to you with the license key.

You must pay for your SLNet license first before a license key will be sent to you.

Enter the Email Address that the license key should be sent to.

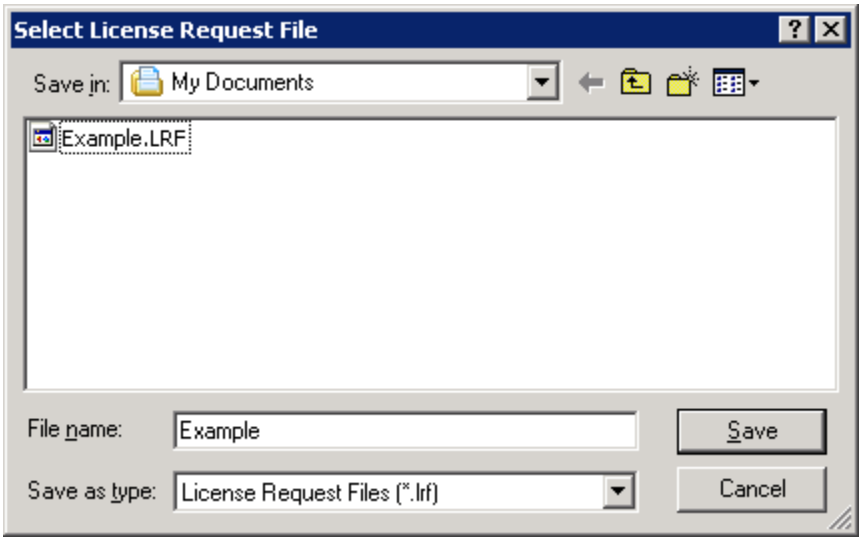
The License Key is based on the **Company Name, Server Location, Server name,** and **User Count**.



Click ***OK*** to save a license request file on this server.

You must pay for your SLNet license first before a license key will be sent to you.

Enter a name for the file, ***Save*** it to this server, and then email that **`.LRF`** file to licensing@seattlelab.com.



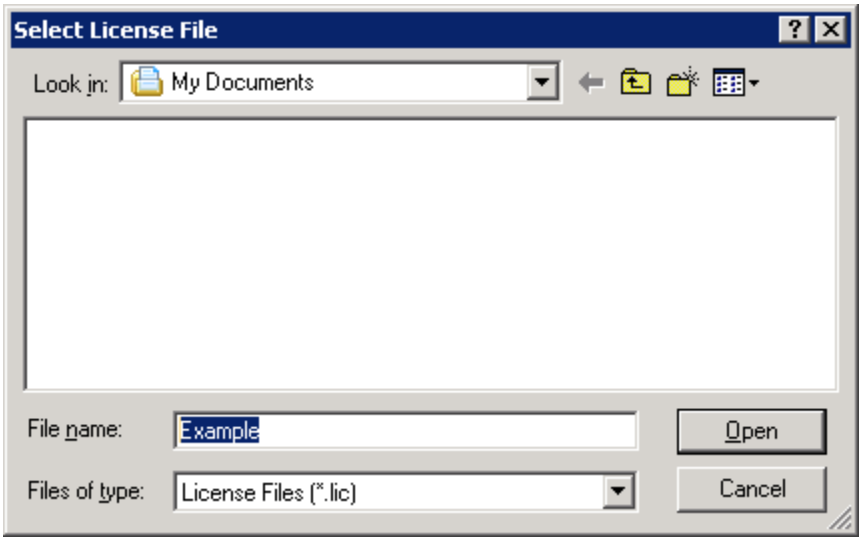
You will receive a **`.LIC`** license file in your email within two business days.

You must pay for your SLNet license first before a license key will be sent to you.

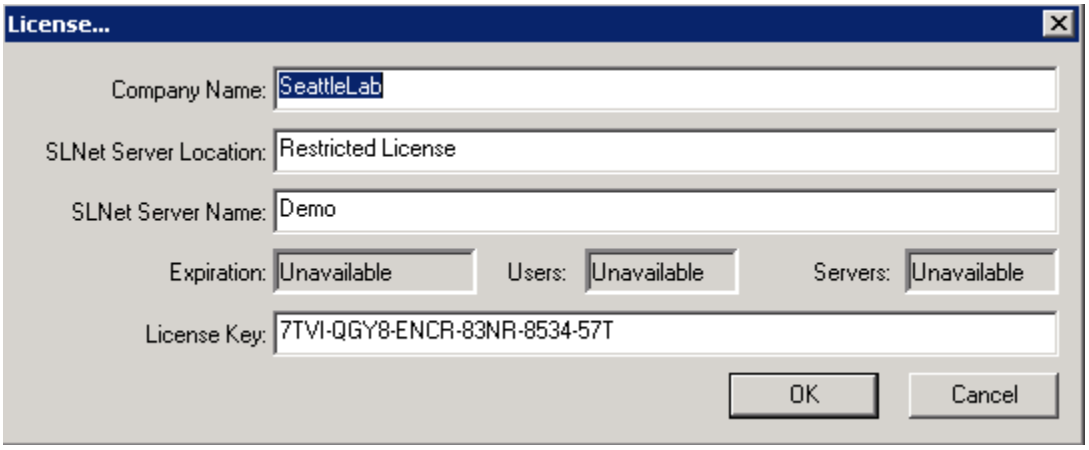
Load a License Key

To load your license file into SLNet, click on the ***Load a License*** button.

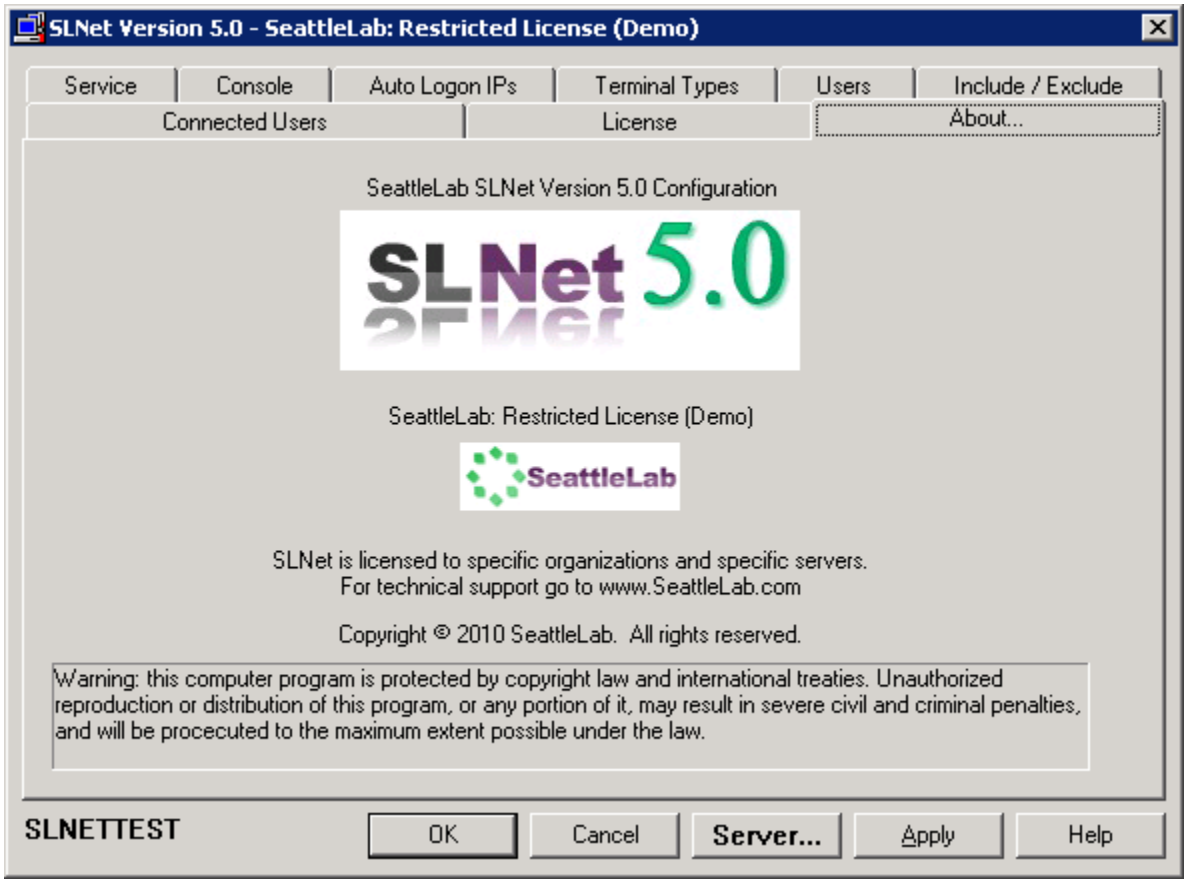
Open the license file (saved from the email you received).



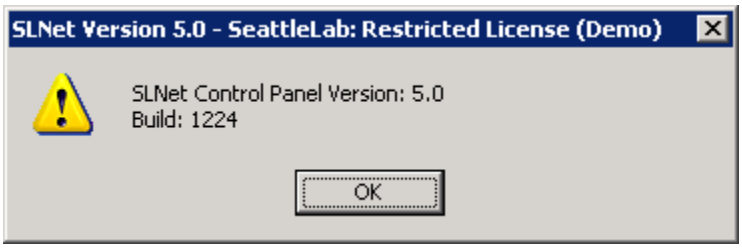
Your license authorization is displayed. Click ***OK***.



About


SLNet Version 5.0 - SeattleLab: Restricted License (Demo)
Service
Console
Auto Logon IPs
Terminal Types
Users
Include / Exclude
Connected Users
License
About...
SeattleLab SLNet Version 5.0 Configuration
SLNet 5.0
SeattleLab: Restricted License (Demo)
SeattleLab
SLNet is licensed to specific organizations and specific servers.
For technical support go to www.SeattleLab.com
Copyright © 2010 SeattleLab. All rights reserved.
Warning: this computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be procecuted to the maximum extent possible under the law.
SLNETTEST
OK
Cancel
Server...
Apply
Help

Click the ***Help*** button to see the version and Build number of your SLNet software.



Logging on to SLNet

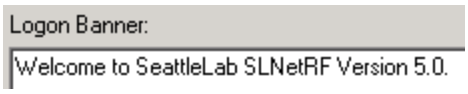
A Telnet client can be any of a variety of devices: a process on a UNIX machine, a Macintosh running MacTCP, a network terminal, a dumb terminal attached to SeattleLab Terminal Server, a Windows terminal emulator, etc.

Many SLNet users connect through handheld terminals to their applications such as warehouse inventory systems, SAP (SLNet usage is recommended in SAP's documentation), etc. These RF devices can emulate a variety of terminal types.

When you initially launch a Telnet client, it is not connected to anything. You need to initiate a Telnet connection. From some Telnet clients, you do so by typing commands at a Telnet command prompt. For example, `connect hostname` or `open hostname` (where hostname is the server name you are connecting to).

Logon Welcome Message

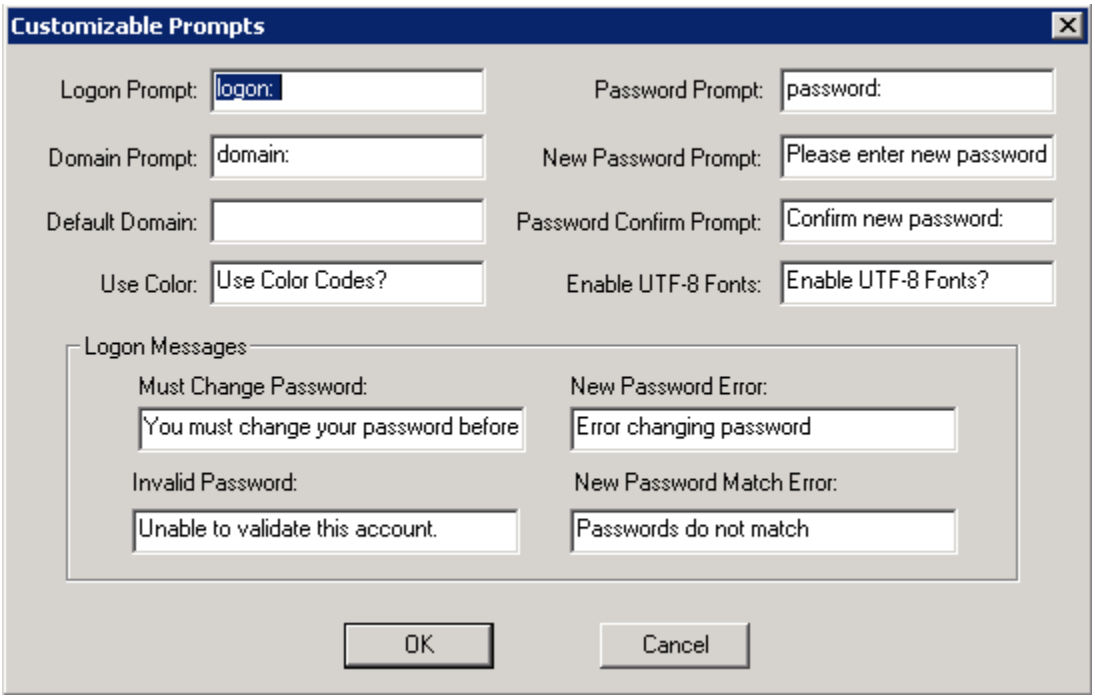
SLNet will display a logon welcome message to the user. This message can be easily customized on the SLNet Service tab.



The user will be prompted for their user name and password.

User Prompts

You can change any of the SLNet user prompts to meet your needs. Click on the Prompts button on the Console tab.



SLNet defaults to the domain name on the Customizable Prompts. If the Default is blank, then the user is prompted for a domain name.

A user can log onto another domain by entering the domain name before the user name at the logon prompt, separated by a back slash (\). For example: `PRODUCTION\MYUSERID`.

When a user has successfully logged on, they will be taken to the shell specified in the Configuration Screen for your user. (The default is the Windows Common Command Shell.)

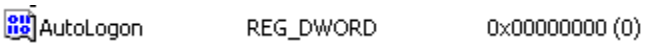
Automatic Logon

With SLNet you can set up an auto logon using a default user account for authentication or by IP address. With auto logon any device connecting to SLNet will automatically be logged on without having to enter a user name or password and will be placed in the shell specified in the User Configuration. The default location is the Windows Common Command Shell.

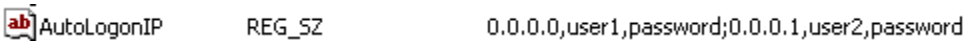
Auto Logon by IP address

You will need to modify a registry value in the root of the SLNet registry *HKEY_LOCAL_MACHINE > SOFTWARE > Seattle Lab > SLNet* called AutoLogon and AutoLogonIP.

Open the Registry Editor through the Start Menu (*Start > Run* and enter `regedit`).



Right click on AutoLogon and change the value to 1.



Right click on AutoLogonIP and change the value to: `IP address, userID, password;` for as many IP addresses as you want to auto logon. *You must separate the string triplets with a semi colon.*

You must restart the SLNet service for the changes to take effect.

Exiting an SLNet session

Your Telnet client will remain connected to SLNet until you disconnect or type **`exit`** at a command prompt.

Automatic Reconnect

SLNet will automatically connect a user back to their application if there is a temporary disconnection (for example, an RF device in a warehouse has its signal disrupted).

SLNet Terminal Emulation

SLNet can be customized to meet any terminal emulation / application character needs.

ANSI terminal emulators lack all the keys on a PC keyboard, yet software written to be run on PCs makes extensive use of these keys. To allow terminal emulator users to simulate the pressing of keys they do not have, SLNet provides mapping sequences. These sequences begin with `CTRL-A` (A for *Attention*). You first press `CTRL-A`, then another key, to simulate the pressing of a single PC keyboard key.

The keys that need to be simulated fall into three basic groups: Cursor keys (Home, Pg Up, etc.); ALT-Keys (keystrokes entered while the ALT key is held down); and PC function keys.

This chart summarizes the `CTRL-A` sequences used to simulate these keys. Where ellipses (...) are used, they indicate that the pattern set by the preceding simulation is followed. For example, to simulate Function Key 2, you would use the same sequence as for Function Key 1, only replacing the 1 with a 2. The sequence must be completed in less than one second, or no key will be simulated, and the sequence that you typed will be sent to the application verbatim.

The Special Keys allow you to control processes you have generated from SLNet. It is possible from SLNet to invoke a process of which you lose control. For example, suppose that from a terminal emulator via SLNet you type `Notepad` at the command line. SLNet will invoke the Notepad applet, which is a Windows program. From your terminal emulator, you have no way of getting input to a Windows program. The Special Keys allow you to terminate this process.

The special keys affect only processes you have created. You cannot use them to list or kill another user's processes

Sequence	Simulates
Cursor Keys	
`CTRL-A i`	Insert
`CTRL-A d`	Delete
`CTRL-A h`	Home
`CTRL-A e`	End
`CTRL-A u`	PgUp
`CTRL-A n`	PgDn
`CTRL-A t`	Tab
ALT Keys	
`CTRL-A A`	ALT-A
`...`	...
`CTRL-A Z`	ALT-Z
Function Keys	
`CTRL-A 1`	F1
`...`	...
`CTRL-A 9`	F9
`CTRL-A 0`	F10
`CTRL-A - (minus)`	F11
`CTRL-A = (equal)`	F12
Sticky Keys (Active until second key pressed)	
`CTRL-A a`	ALT
`CTRL-A c`	CTRL
`CTRL-A s`	SHIFT – Caps Lock On
Special Keys	
`CTRL-A p`	List Processes
`CTRL-A k`	Close last application on list
`CTRL-R`	Refresh Screen
`CTRL-A SPACE`	Refresh Screen
`CTRL-A l`	Forced exit of application
`CTRL A CTRL K`	Keyboard analysis – key ASCII codes are displayed
`CTRL A CTRL R`	Raw keyboard input. Keys are not translated to Windows keys.

SLNet Logging

SLNet keeps a record of its activities by entering messages in the Windows Event Log as events occur.

To view event logs for SLNet, go to the Log menu on the tool bar and select *Application.*

You may select which types of events you wish recorded in the Event Log by going to the SLNet Configuration panel *Service* tab. Check Logging - set Logging On and adjust the Options.

Logging Message examples

The [%1] and [%2] tags in the following examples are values that are filled in by SLNet at the time the event is generated

- If the socket SLNet is trying to use is already in use by another program the user will get an Error message. The number *1000* will appear in the Event column and the Error message will be: “The [%1] service could not be started.”
- If a user who does not have permission to use SLNet attempts to log on, the number *2000* will appear in the Event column, and the Warning message will be: “User [%1] attempted to log on with no profile. There was no default.”
- When a user logs onto SLNet, the number *2002* will appear in the Event column and the Information message for that event is: “Accepting connection from IP: [%1]. Connection Number [%2].”
- When the Telnet service is successfully started, the number *3000* will appear in the Event column and the Success message for that event is: “[%1] has been successfully started.”

If the user is connecting via a proxy server, the IP address will always be the address of the proxy server rather than the client machine on the other side of the proxy server.

SLNet Error Codes

The [%1] and [%2] tags in the Error Message column are values that are filled in by SLNet at the time the event is generated. Generally speaking, [%1] tags represent applications and [%2] tags represent machine IP addresses.

Please contact our technical support team for assistance. Most error codes are from authentication security.

Event	Type	Error Message
1000	ERROR	The %1 service could not be started.
1001	ERROR	Unable to find socket library.
1002	ERROR	Unable to allocate memory for Socket Info Structure.
1003	ERROR	An error occurred while attempting to allocate memory.
1004	ERROR	%1 has not been installed correctly. Please re-install.
1005	ERROR	%1 %2 (generic)
1006	ERROR	%1
1007	ERROR	The user's process returned %1.
1008	ERROR	Assert in file %1.
1009	ERROR	Invalid registration data. The service is shutting down.
1010	ERROR	The demo version of %1 has expired.
10000	ERROR	The Socket error is undefined.
10004	ERROR	The Socket has been interrupted.
10024	ERROR	No more file descriptors are available.
10036	ERROR	A blocking Windows Sockets operation is in progress.
10041	ERROR	The specified protocol is the wrong type for this socket.
10043	ERROR	The specified protocol is not supported.
10044	ERROR	The specified socket type is not supported in this address family.
10047	ERROR	The specified address family is not supported.
10050	ERROR	The Windows Sockets implementation has detected that the network subsystem has failed.
10052	ERROR	The network has been reset.
10053	ERROR	The connection was aborted.
10054	ERROR	The connection was reset.
10055	ERROR	No buffer space is available. The socket cannot be created.
10060	ERROR	A timeout has occurred.
10093	ERROR	A successful WSAStartup must occur before using this API.
2000	WARNING	User %1 attempted to logon with no logon profile. There was no default.
2000	INFORMATION	The terminal type is: %1.
2001	ERROR	A connection from address %1 was rejected because it is not in the accepted address range.
2002	INFORMATION	Accepting connection from IP: %1. Connection Number %2.
2003	INFORMATION	IP %1 Disconnected.
2004	INFORMATION	The demonstration period will expire on %1.
2005	INFORMATION	%1 user version initialized.
3000	SUCCESS	%1 has been successfully started.
3001	SUCCESS	%1 has been successfully stopped.

Appendix A. Troubleshooting

SLNet is designed to be a powerful solution to Telnet needs.

Problems usually occur when you have made a change in your environment. This may be an operating service pack, a security change, new device, or changing users.

Before You Contact SLNet Technical Support

In order to do our best for you, we require accurate and timely information in the formats we request. Our records show that over 50% of all incidents reported are resolved as permission problems or not following the directions in the documentation.

The Windows Event log has many clues to resolving SLNet issues.

Here are steps to follow before you call our technical support team:

1. Review the Windows Event log. It is a good source for error messages on security and connectivity.
2. Validate that a user has a good connection to your network and a valid Windows account.
3. Verify that the user has permission to log on locally to the server that SLNet is installed on.
4. Turn on SLNet logging for a user and review the log details. Often you will find the user has entered invalid characters or data.

When you contact SLNet technical support, please let us know your Windows operating system release and hardware environment, SLNet version, and a complete description of your question or problem (let us know if you can recreate it), and have access to the SLNet server.

SeattleLab Professional Services

If you need special assistance with customization of SLNet, please contact our technical support group to arrange for billable professional services time.

Contacting SLNet Specialists

Customers with active maintenance support contracts have access to technical specialists.

Email Technical Support: support@SeattleLab.com

Telephone numbers and hours of business for our offices are available on our web site - **www.SeattleLab.com**.

We'll "do what it takes" to assist you with your questions or problems.

Appendix B. Customization

Named Pipes

SLNet works with unmodified off-the-shelf character applications by reading the console output buffer at periodic intervals, noting any changes made in the buffer, translating those changes into a character stream, and directing that character stream to the Telnet socket. The character stream contains ANSI escape sequences for cursor control to correctly format the screen. This design allows SLNet to support the widest possible range of third-party products, without requiring customers to make any modification to their software.

For customers who have the ability to modify their software, however, or who are writing new software, SLNet provides a more direct output connection. This connection takes the form of a Named Pipe. There are two advantages to using direct output to the named pipe:

1. It is more efficient, allowing given hardware to support more processes in addition to allowing for each process to complete more quickly.
2. It allows support for non-ANSI terminal emulations. The SLNet Named Pipe can be handled in the following way:

   When SLNet creates a process, it places the name of the pipe it creates for that process in a variable in the process's environment. The Pipe Name field defines the name of this environment variable. To read this environment variable from C code, a program could call the `getenv` function. The name received from the environment can then be used in a call to the Win32 API `CreateFile` (which is used to open files and other devices as well as to create them).

   If your application shells out to run other applications that do not know about the Named Pipe, you should close the pipe, and SLNet will return to screen-scraping mode. When your application regains control, it can then re-open the pipe. The Win32 API `CloseHandle` is used to close the pipe:

   Characters may be directed to the named pipe by calling the Win32 API function `WriteFile`.

The following C code sample demonstrates how to open the SLNet Named Pipe directly.

```
#include <fcntl.h>  /* Needed only for _O_WRONLY definition */ #include <io.h>
#include <stdlib.h> #include <stdio.h> #include <string.h>

void main (void)
{ char PipeName[256]; char *EnvPipeName; char Temp[256]; int PipeHandle; int
BytesRead;

      EnvPipeName = getenv ("PIPENAME");
      /* ** the pipename on NT is \\.\pipe\<pipename> where <pipename> is the
      value stored **in the PIPENAME environment variable set by SLNet */

      if (EnvPipeName){ sprintf (PipeName, "\\\\.\\PIPE\\%s", EnvPipeName);
      PipeHandle = open (PipeName, _O_WRONLY);

      if (PipeHandle == -1) {
            printf ("Error: Unable to open the named pipe %s.\n", PipeName);
            exit (-1); }

sprintf (Temp, "This program now controls the output to the terminal\n Type
'exit' to end.\n");
```

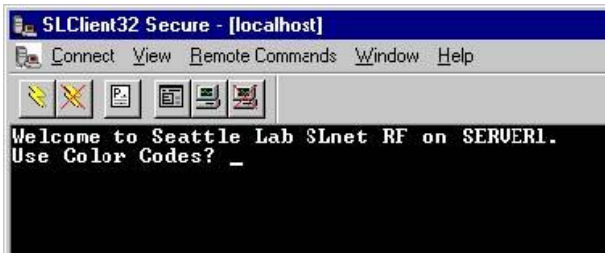
```
      write (PipeHandle, Temp, strlen (Temp));
      while ((BytesRead = read (0, Temp, 256))) {
      /* File handle 0 is standard in or the keyboard */
      if (BytesRead == -1) /* an error occurred */ break;

            if (strnicmp (Temp, "exit", 4) == 0) /* The user chose to exit */
            break;
      write (PipeHandle, Temp, BytesRead); } close (PipeHandle);


} else {
printf ("SLNet is not running\n"); }
exit (0);
}
```

Appendix C. Miscellaneous

SLclient32


SLClient32 Secure - [localhost]
Connect View Remote Commands Window Help
Welcome to Seattle Lab SLnet RF on SERVER1.
Use Color Codes? _

SLclient is a fully featured, secure 32-bit Telnet client for Windows only. It is available from the Start menu `Start > Programs > SL Products > SLNet >`

Secure Logon Capabilities

SLclient32 provides secure logon for SLNet users. When a user enters their user ID and password, SLclient32 encrypts them and sends them to SLNet where they are decrypted. This allows complete security when sending IDs and passwords over the Internet.

In a standard dialup Internet connection using Point-to-Point Protocol (PPP), the data is encrypted with a Point-to-Point Tunneling Protocol (PPTP) wrapper. This encryption is a 40-bit international or standard process. Together SLNet and SLclient32 use a stronger encryption when sending logon data; the encryption is used only for authentication, however, not general data security.

Note: SLclient32 only works with SLNet. It will not work with any other product.

Appendix D. Multiple Console Windows

SLNet allows you to remotely open multiple console windows on the Windows desktop just as you would if you were sitting at the terminal and needed to run some programs simultaneously. This functionality is designed to be reasonably unobtrusive, displaying a menu of the processes as needed, otherwise remaining completely invisible as you work in the active window.

A single console window requires much less bandwidth and supports far more concurrent connections than multiple window do, so it is recommended that using multiple console windows (by using SLCons) only be done for administrative purposes.

Select Console Type

SLCons and SLCon

SLNet allows you to use:

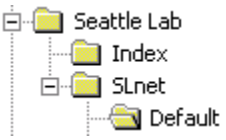
- `SLcons.exe`, which has more functionality but supports fewer concurrent connections or
- `SLcon.exe`, which has less functionality but supports more concurrent connections.

For example, an administrator may want to log on with **SLCons** due to its increased functionality and limited connections while users using a hand held device may want to use **SLCon** because they may have more connections and do not need all of the functionality.

SLNet allows you to select the console type on a per-user basis.

You will need to create and modify a registry value in the root of the SLNet registry *HKEY_LOCAL_MACHINE > SOFTWARE > Seattle Lab > SLNet > UserID* called Console.

Open the Registry Editor through the Start Menu (*Start > Run* and enter `regedit`).



Select the user ID to modify.

Right click in an open space in the right panel and select New String Value.

Type Console for the String name.

Right click on Console and change the value to:

- `path to SLCon.exe;7 or`
- `path to SLCons.exe;9`

You must restart the SLNet service for the changes to take effect.

Working with Multiple Console Windows

When a user logs on, SLNet creates a console window for the default shell, normally the Windows Command Shell (this can be changed in SLNet Control Panel applet for any user). The user can work in this console window as they would in any Telnet session or they can immediately start the next process in a second console window.

To start a second process, type the command to start that process, for example: **`start cmd`**. On the Windows server there are now two console windows available to the user but they are still looking at the Telnet client window displaying the contents of the first console window. Their keyboard input will continue to go to this first window (the second window is not displayed until they switch to it).

To switch to the second console window, press `CTRL-A 2` (hold down the `CTRL` key and press `A`, release `CTRL` and `A`, and then press `2)`. By default, you have two seconds to send the `2` keystroke after pressing the `CTRL-A` combination, otherwise `^a` will be sent through verbatim. This timeout is configurable in the registry.

Proceed to work in this second window, just as you would in any normal Telnet session, by typing the commands you want to execute. SLNet forwards all keystrokes to your current console window and will forward that window's output to your terminal. The first window still contains any process you started there, but until you switch back to it, you will only see the work in the second console window.

The status line at the bottom of the screen confirms which console window is displayed as well as occasionally providing other information. The status message is displayed for 5 seconds and then disappears.

The following SLNet commands are available to you while working in the console window you have created, in addition to commands available in the shell that is running:

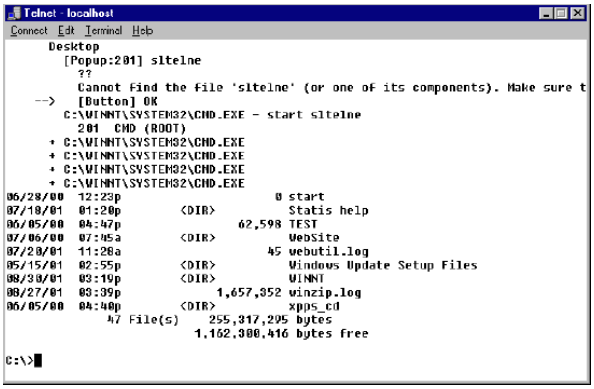
Command	Result
`tlist`	lists the tasks with their process Ids
`kill <PID>`	kills the specified process

SLNet also provides Hot Key commands to work in a console window and to switch between windows. This is an abbreviated list of key sequences that are useful for working in a console window "normal mode." A second set of key sequences is available when working in "menu mode".

Using the Menu

To gain an overview of the windows and processes you are running in the Telnet client window you can display the Menu at any time by pressing `CTRL-A w`.

The Menu is displayed in the first few lines at the top of your Telnet client window and the current console window occupies the rest of the window below the Menu.



The top line of the Menu indicates the Desktop that contains the windows that are currently running.

- The arrow (**- - >**) highlights the currently selected menu item.
- The plus sign (**+**) indicates that collapsed sub-tree information is available under the menu item.
- Popups (dialog boxes or error messages) are displayed directly under Desktop unless the view has been changed.
- When the information is expanded, processes are shown under the windows they are running in with the Process ID followed by the executable name for process then the current window title for windows and console groups.
- `XXX` next to a window or process indicates it has exited or been killed, it will disappear from the list shortly after you move the highlight arrow.

The active window may be displayed in a different color (depending on your client, selected terminal type, and the terminal databases description of that terminal type).

While the Menu is displayed you are no longer in any of your console windows and all the input from the keyboard is used to navigate within the Menu. Use the arrow keys on your keyboard if available or the keystrokes in the table below to: move selection arrow; expand or collapse information; locate popups; kill processes, popups, or windows; and switch to a selected window.

This key…	has this effect when the Menu is displayed
`j`	Moves highlight arrow down one line
`k`	Moves highlight arrow up one line
`b`	Back one page (PgUp)
`f`	Forward one page (PgDn)
`+`	Shows sub-items hidden under the highlighted item
`-`	Hides sub-items
`TAB`	toggles collapse/expand display for selected menu item
`Z` (capital Z)	kills highlighted process or popup
`ENTER` (when pointing to a window)	closes the menu and makes the highlighted window the current console window
`ENTER` (when pointing at a popup	locates popup under the process that created it
`v` (when pointing at a popup)	displays popup location
`ESC`	hides the menu and returns to selected console window
`q`	hides the menu and returns to selected console window

Initially, all console windows are collapsed, so that the desktop, popups, and windows are visible, but process sub trees are not. When a console window is expanded, each process tree associated with that console is listed by ascending Process ID (note that processes form a hierarchy, and can appear under other processes).

Locating and dispatching Popups

The work you do in your console window may cause an error message or dialog box requesting some response and these will appear as a popup in your session, preventing further work in that console window until the popup containing that error message or dialog box has been dispatched.

If your work in a session has been interrupted and you can no longer input to that window, it is best to check for a popup by bringing up the Menu (press `CTRL a CTRL w`).

If the Menu lists a popup under the desktop, use your arrow keys or equivalent keystroke combination to navigate to it. Press `ENTER` or `v` to view the popup under the associated console window. The Process Identifier (shown inside the bracket) refers to the process that created the popup.

The safest way to dispatch the popup is to move the highlight to the popup on the menu and press `Z` to kill it. If that fails, the second safest thing to do is to kill the console window. If you don't want to kill the window, go to the process associated with the popup and kill the process directly using `Z`.

Applying `Z` to a process is more dangerous than killing a window because the process won't have a chance to exit gracefully. Since the menu is updated every few seconds, it may take a few seconds for windows that you kill to disappear from the menu.

Killing a window or dialog box using `Z` is the same as clicking the "*Close*" button in the upper right corner of the window.

Hot Key Commands

** Where ellipses (…) are used, they indicate that the pattern set by the preceding simulation is followed. For example, to simulate Function Key 2, you would use the same sequence as for Function Key 1, only replacing the 1 with a 2.*

*** These commands affect when the SHIFT key presses and key releases are sent on to the window. They do not affect programs operating in pass through modes, as we send these programs ASCII streams of characters, rather than keystrokes.*

**** WARNING: This command is bound to a shifted key. Windows' Telnet client will appear to HANG if you press CTRL+SHIFT+<x>, where <x> is any key. Be sure to release CTRL (and a) before typing the next character.*

Sequence*	**Result when working in Menus**
`j`	Move selection arrow down one line
`k`	Move selection arrow up one line
`b`	Move back one page (PgUp)
`f`	Move forward one page (PgDn)
`+`	Expand to view processes under the selected window
`–`	Collapse information under the selected window
`TAB`	Toggle collapse/expand display for selected window
`BACKSPACE`	Toggle collapse/expand display for selected window
`Z (capital Z)`	Kill a listed window or process or popup
`ENTER` (When pointing to a window)	close the menu and make the highlighted window the current console window
`ENTER` (When pointing at a popup)	locate popup under the process it is associated with
`v` (When pointing at a popup)	display popup location
`ESC`	Hide the menu and returns to selected window
`q`	Hide the menu and returns to selected window
`CTRL-a CTRL-k u`	Up

CTRL-a CTRL-k d	Down
CTRL-a CTRL-k r	Right
CTRL-a CTRL-k l	Left
CTRL-a CTRL-k p	PgUp
CTRL-a CTRL-k n	PgDn
CTRL-a h or CTRL-a H	Display Online help menu (use menu keystrokes to navigate in menu)
CTRL-a CTRL-h	Display Online help menu (use menu keystrokes to navigate in menu)
CTRL-a ? or CTRL-a /	Display Online help menu (use menu keystrokes to navigate in menu)
CTRL-a CTRL-c	Create another login process (using the same shell and parameters)
CTRL-a c	Create another login process (using the same shell and parameters)
CTRL-a 1	Switch to the first open window
...	Switch to windows 2-9
CTRL-a 0	Switch to the 10th open window
CTRL-a	Switch to the 11th open window
CTRL-a =	Switch to the 12th open window
CTRL-a CTRL-w	Display the Menu (“Windows”)
CTRL-a w	Display the Menu (“Windows”)
CTRL-a CTRL-d	Display the Menu (“Processes”)
CTRL-a p	Display the Menu (“Desktop”)
CTRL-a B	Sends CTRL+BREAK to the window***
CTRL-a C (capital C)	Sends CTRL+C to the window***
CTRL-a #	Disconnect***
CTRL-a CTRL-r	Repaint screen
CTRL-a SPACE	Repaint screen
CTRL-a CTRL-l	Repaint screen
CTRL-a CTRL-k 4	Left
CTRL-a CTRL-k 6	Right
CTRL-a CTRL-k 7	Home
CTRL-a CTRL-k 8	Up
CTRL-a CTRL-k 9	PgUp
CTRL-a CTRL-k 0	Insert
CTRL-a CTRL-k .	Delete
CTRL-a CTRL-k t	Backtab
CTRL-a CTRL-k u	Up
CTRL-a CTRL-k d	Down

CTRL-a CTRL-k l	Left
CTRL-a CTRL-k r	Right
CTRL-a CTRL-k i	Insert
CTRL-a CTRL-k h	Home
CTRL-a CTRL-k e	End
CTRL-a CTRL-k p	PgUp (Previous Page)
CTRL-a CTRL-k n	PgDn (Next Page)
Function Keys	
CTRL-a f 1	F1
...	F2, F3, F4…
CTRL-a f 9	F9
CTRL-a f 0	F10
CTRL-a f -(minus)	F11
CTRL-a f = (equal)	F12
Shift Next Character**	
CTRL-a CTRL-x a <char>	ALT <char> (where <char> represents the character that Alt was to be applied to. EXAMPLE: for ALT F, type CTRL a CTRL x a f)
CTRL-a CTRL-x s <char>	SHIFT <char> (where <char> represents the character that SHIFT was to be applied to. EXAMPLE: for SHIFT F, type CTRL a CTRL x s f)
CTRL-a CTRL-x c <char>	CTRL <char>(where <char> represents the character that CTRL was to be applied to. EXAMPLE: for CTRL c, type CTRL a CTRL x c c)
^a^x^c ^a^x^a ^a^k.	CTRL+ALT+DEL If you set a temporary shift for Ctrl (^a^x^c), then set another temporary shift for Alt, both shifts will be pending for the next key DEL off the keypad (^a^k).

This page intentionally left blank.